

暦に関する Web API – 暦法の変換と期間の計算

関野 樹 (総合地球環境学研究所)

暦に関する情報処理のため、以下の特徴を持つ Web API が実装された。1) 暦日だけでなく、年号、暦年、暦月など (以下、まとめて「暦日等」) の処理が可能である、2) 暦法の変換だけでなく、暦日等への期間の加算、2つの暦日等間の期間長や相対関係を取得する機能を実装している、3) セマンティック Web 技術や Linked Data との高い親和性を有する。本 Web API により、さまざまな時間情報が暦法の壁を越えて扱えるようになり、新たなコンテンツやアプリケーションの開発が進むと期待される。

Calendar Web API – Calendar conversion and calculation of duration

Tatsuki Sekino (Research Institute for Humanity and Nature)

A Web API has been developed in order to operate calendrical information. The Web API has following features: 1) Calendar months, years and eras are available as well as calendar dates, 2) In addition to feature of calendar conversion, features of adding duration to calendrical periods and calculating durations and relative positions between calendrical periods, are implemented, 3) there is high affinity with semantic web and linked data technologies. It is expected that novel web contents and application, in which temporal data with different calendars are available, are developed, using the Web API.

1. まえがき

時間は、情報の可視化や解析において、データを連携させるための接点の一つであり、並べ替えや長さの比較のための基準軸でもある。また、検索においても、データの特定や検索対象の絞り込みに頻繁に利用される。

ところが、実際のデータでの時間情報の扱いは、必ずしも容易ではない。日付は、和暦やグレゴリオ暦 (いわゆる「西暦」) などのさまざまな暦法に基づいている。このため、異なる暦法に基づく日付の間で前後関係や長さの比較をするには、いずれかの暦法へ日付を正規化する必要がある。また和暦では、異なる年号を持つ日付の前後関係が容易には分からない。このため、和暦の日付をいったん西暦の日付に変換した上で、解析や可視化などの操作が行われることが多い。

日付の暦法を変換する機能は Web 上で数多く提供されてきた[1, 2, 3]。これにより、以前は対応表などを用いて手作業で行われてきた暦法変換が Web ページ上で手軽に行えるようになった。しかしながら、実際に要求されている操作は上述のように解析や検索のための前後関係や期間の長さの比較であり、暦法の変換は、やむを得ず行われる中間処理に過ぎない。つまり、和暦など元の暦法のままこれらの操作が可能であれば、和暦の日付をわざわざ西暦に変換する必要はないのである。

これは、ユーザの利便性とどまらず、処理の正確性にも関わる問題である。「寛文7年から元禄14年」を西暦に変換して「1667年から1701

年」として検索を行ったとする。しかしながら、厳密には、寛文7年は1667年1月24日～1668年2月11日であり、元禄14年は1701年2月8日～1702年1月27日であって、和暦年と西暦年の期間は一致しない。したがって、「寛文7年から元禄14年」を条件とした検索のつもりで「1667年から1701年」で検索を行えば、誤った時間範囲を検索することになる。もし、検索する時間範囲として直接和暦の年を指定することができれば、こういった暦法変換に伴う問題も解決されるはずである。

検索での時間範囲の指定には、日付の前後関係を得る仕組みが必要になる。ところが、前述のとおり暦法変換のサービスは数多く提供されているものの、日付の前後関係や期間の長さの比較するサービスは極めて限定的である。一般的な表計算ソフトにはこのような期間の計算に関する機能を持つものもあるし、Web フォーム上で同様の機能を提供している例もある[3]。ところが、その多くは、現代の、しかも西暦に限定されており、近世以前の国内の史料で使われた和暦の日付でこういった操作を行うことは難しい。

さらに、暦に関する機能が Web フォーム上でのサービスでしか提供されていないことも問題である。可視化、解析、および検索など、何らかの操作でこれら機能を活用しようとするれば、Web API (Application Program Interface) として外部から機能そのものを利用できなければならない。ところが、期間の計算に関する機能はもちろんのこと、暦法変換の機能さえも Web API として提供されているものが無い。このため、暦法変換や期

間計算などの機能をその都度改めて構築することになり、日付を使った新たなコンテンツやアプリケーション開発の障害となっている。

本研究は、HuTime プロジェクトで提供している、日付文字列の解釈[4]や、暦に関する Linked Data [5, 6]の機能を活用し、暦法の変換や期間の計算を行うための Web API (以下、「HuTime Web API」) を新たに提供しようとするものである。これにより、データベースの検索において、和暦を直接入力して検索対象を指定したり、Web ページ上で、和暦の日付に自動的に西暦の日付を付加したりといった、新たな機能の実装が容易になる。この結果、暦法の壁を越えて時間情報の処理を行うコンテンツやアプリケーションの開発が加速すると期待される。

2. 機能の概要

HuTime Web API には、以下の 5 つの機能が実現された。特に、(3)～(5)の機能は、HuTime プロジェクトで提供されているサービスも含め、これまで Web API としては提供されてこなかった機能である。

なおここでは、暦法に基づいて規定された年号、暦年、暦月、および暦日などの時間範囲を「暦日等」と総称する。これらは、暦に関する Linked Data の暦リソースと同等である。「日付」は、これらの暦日等を表す文字列であり、暦リソースのラベルは、日付の一つとして位置づけられる。

(1) 暦日等に関する情報の取得

暦日等に関して、暦日等を表す文字列(日付)、時間範囲、期間の長さ、前後関係や階層関係、閏の有無などの情報を取得する。これらは、暦に関する Linked Data [5, 6]で提供されている情報と整合性を持つ。なお日付は、HuTime 暦変換サービス[4]と同じ方法で書式の指定が可能である。

(2) 暦法の変換

暦日等を異なる暦法の暦日等に変換する。HuTime Web API では、暦日だけでなく、年号、暦年、暦月といったさまざまな種類の暦日等の暦法変換も可能である。しかしながら、暦年や暦月の決め方は暦法ごとに異なるため、同じ時間範囲を持つ変換先がない場合がある。このため、変換先が変換元の時間範囲の始点、終点のいずれを基準にして変換するかを選択できるようになっている。また、変換元と変換先の暦日等の種類が異なる暦法変換もできる。したがって、「和暦の年号の始点と終点に相当する西暦の暦日」を求めるといった操作も可能である。

(3) 暦日等への期間の加算

暦日等に対して、日数、または、年数+月数+日数で表された期間を加算し、その暦日等を得る。年数と月数は、これらの根拠となる暦法の指定が

可能であり、「元禄 14 年 3 月 14 日の 22 か月後」を、閏月を含む当時の和暦に基づいて計算できる。なお、負の値を指定することで、「22 か月前」といった過去に遡る計算も可能である。

(4) 暦日等間の期間の長さを算出

2 つの暦日等間の期間の長さを算出する。期間の長さは、日数、年数+月数+日数、年数+日数、月数+日数の表現から選択できる。また(3)と同様に、年数と月数を計算する根拠となる暦法を指定できるため、「寛文 7 年 8 月 11 日から元禄 14 年 3 月 14 日まで、和暦で数えて何年何か月何日間あるか」といった計算も容易に実施できる。

(5) 暦日等同士の間接関係を取得

2 つの暦日等について、時間軸上での相対関係を得る。結果は、Allen's Time Interval Algebra [6]に基づいて生成される。Allen's Time Interval Algebra は、2 つの時間範囲がとり得る関係を 13 通りに分類したもので(図 1)、時間情報を扱う上での基礎理論として広く普及している。実際に Allen's Time Interval Algebra は、地理情報[8]や時間メタデータ言語(Time ML) [9]などの国際規格、W3C が策定した時間オントロジ[10]でも採用されており、時間範囲の相対関係を表す標準となっている。

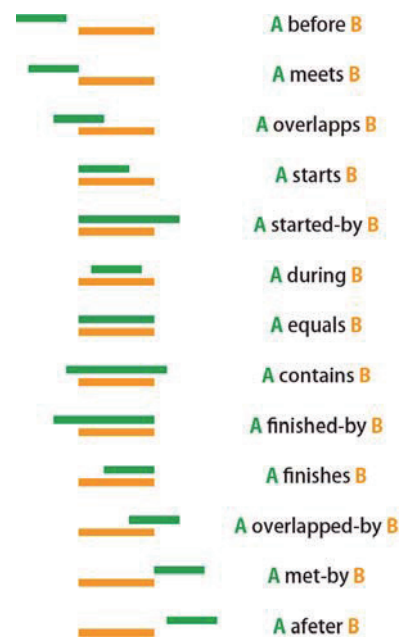


図 1 2 つの時間範囲がとり得る相対関係を 13 通りに分類した Allen's Time Interval Algebra [6].

Figure 1 Allen's Time Interval Algebra which is a classification of possible relative relationship between temporal intervals [6].

この Allen's Time Interval Algebra で表される相対関係を検索に応用すれば、「寛文 7 年の間に終

わった」「元禄年間をまたぐ」「元禄 14 年の間に始まった」といった、これまでにない柔軟な検索条件の指定も可能になる。

3. HuTime Web API の利用

HuTime Web API の利用マニュアルは、以下より入手可能である。

<http://ap.hutime.org/cal/index.html>

利用方法の詳細はこの Web マニュアルに譲り、本稿では、HuTime Web API に特徴的な機能とそのねらいを解説する。

3-1. API へのアクセス

HuTime Web API は、RPC (Remote Procedure Call) 型のインタフェースである。操作に必要な変数に値を設定し、POST または GET メソッドで以下の URL に要求する。

<http://ap.hutime.org/cal/>

不適切な値の設定などによりエラーが発生した場合は、HTTP ステータスコードにより、該当するエラーが返される。

また、HuTime Web API は、JSON-RPC 2.0 [11] にも対応する。これにより、JSON 形式のデータを扱える JavaScript などでの Web アプリケーションやコンテンツの開発において、HuTime Web API が容易に利用できるようになる。

3-2. 暦日等の指定

操作の対象となる暦日等は、暦リソースの URI で指定する方法と、必要な値を変数に設定して指定する方法がある。前者は、Linked Data などに由来する RDF の処理に対応するためのものである。一方、後者は、Web フォームなどの RDF を用いないアプリケーションに対応するためのものである。

暦日等は、暦法、暦日等の種類、および暦日等の範囲を示す値の 3 つの要素により特定される。このうち、暦法 ID [5] で指定される暦法、および“date”などの文字列で指定される暦日等の種類 [6] は、暦リソース、変数指定のいずれの方法でも同じである。一方、暦日等の範囲を示す値は、変数指定の場合のみ書式指定が可能で、西暦を伴う和暦の日付 (例：元禄 14 (1701) 年 3 月 14 日) などを入力データとすることができる。

3-3. 暦日等の一括処理

Web フォームを使った暦変換のサービスでは、日付を構成する年号、年、月、日をそれぞれ別々のテキストボックスに入力し、1 件ずつ処理することが一般的である。一方、HuTime Web API では、暦日等を表す日付を解釈する機能により、日付の中の年号、年、月、日を認識できる。このため、改行 (CRLF) により区切られた複数の日付

が与えられた場合には、これらを一括で処理し、同様に改行で区切られた結果を出力する。入力データにエラーがあっても、入力データと出力データが常に行単位で対応するようになっているため、表形式のデータを作成する際に、入出力データを並べる操作を行っても、行にズレが生じない。

複数の暦日等を一括処理することは、利用者にとっての利便性だけでなく、性能面にも関係する。一般的に Web API では、処理時間そのものよりも、ネットワーク上の遅延がスループットに大きく影響する。このため、複数の暦日等を一括で処理すれば、ネットワークトラフィックの削減と処理全体のスループットの改善が期待される。

3-4. 計算に用いる端点

期間の計算では、暦日等の始点、終点のどちらを基準にするかが問題となる。たとえば、「11 月 7 日の 2 日後」は、一般的には 11 月 9 日を指している。日付の 7 に 2 を加算したわけである。この場合、起点の暦日 (7 日) と対象点の暦日 (11 日) の端点は同じであり、始点から始点、または、終点から終点が 2 日間である。ところが、あるイベントの開催期間が「11 月 7~9 日」と示されていれば、開催期間は 3 日間と考えるのが普通である。この場合、3 日間を測る端点が起点 (7 日) と対象点 (9 日) で異なっており、7 日の始点 (0 時) から 9 日の終点 (24 時) までが 3 日間であるという計算である。HuTime Web API では、さまざまな文脈に応じた計算に対応するため、期間に関する処理において、計算に用いる端点を指定できるようになっている。

3-5. 出力形式

HuTime Web API での処理結果の出力形式は、テキストと、RDF に大別される。前者は、処理結果がそのままテキストとして出力されるもので、Web フォームなどでの利用や、CSV 形式などのテキストファイルを取得する操作を想定している。後者は、セマンティック Web 技術を応用したより複雑な処理での利用を想定しており、XML、Turtle、JSON 等の主要なファイルフォーマットを選択できる。

テキストの出力は、暦日等に関する情報を柔軟に取得できることが特徴である。表 1 は暦日等に関する情報として出力できる項目であり、複数を指定することも可能である。

暦日等を表す文字列 (= 日付) を得る操作には、指定子 “text” を使用する。ここで得られる日付は、既存の HuTime 暦変換サービス [4] のように書式を指定できる。また、この日付の出力において、指定した情報が再び暦日等である場合、これについても情報を得ることができる。例えば、指定子 “next” を使って暦日の次 (つまり、翌日) を得る場合、これもまた暦日等であるので、この日付を書式指定して出力するといった操作も可能で

ある（この場合、指定子は“next.text”となる）。指定子を繋ぐことで、より複雑な指定も可能であるが（例：“ofYear.next.uri” - 属する暦年の翌年の暦リソース URI）、サーバへの過度な負荷を避けるため、繋げられる指定子の数は制限されている。

このような出力は、処理の結果が暦日等となる処理、つまり、（1）暦日等に関する情報の取得、（2）暦法の変換、（3）暦日等への期間の加算で可能である。例えば、「和暦の暦日を西暦の暦日に変換し、その暦日を示す暦リソースの URI を得る」といった処理が想定される。

表 1 暦日等に関する情報の項目一覧。出力項目が暦日等である場合は、“.”で繋げるにより、さらにその暦日等に関する情報の出力項目を指定できる。

（例：“next.uri” - 次の暦日等の暦リソースの URI）

Table 1 Output items about a calendrical period. When a specified item is again in a calendrical period, it is possible to specify an output item of the calendrical period, connecting the specifiers with “.” (e.g. “next.uri” - URI of the next resource)

指定子	内容
begin	始点を表す暦日等
end	終点を表す暦日等
previous	前の暦日等
next	次の暦日等
ofMonth	属する暦月
ofYear	属する暦年
ofEra	属する年号
text	暦日等を表す文字列（oform 変数により書式指定が可能）
uri	暦日等の暦リソースの URI
jdBegin	始点のユリウス通日
jdEnd	終点のユリウス通日
dayDuration	日数で表した期間長
calendarDuration	年数+月数+日数で表した期間長
dayOfYear	年のはじめからの日数
isLeap	閏であるかの正否
label	暦リソースのラベル

RDF による出力では、要求された処理内容も含めた結果が RDF で表現される。図 2 は、期間に関する計算の結果として出力される RDF の例である。期間の始点と終点の暦日等は暦リソースで表現されている。これらの暦日等に関する情報を得るには、Linked Data の仕組みなどを活用して、各暦リソースに関する RDF を取得する。期間は空白ノードで表現されており、始点と終点の

暦リソース、および、期間の長さ（日数）がそれぞれ RDF で関連付けられている。

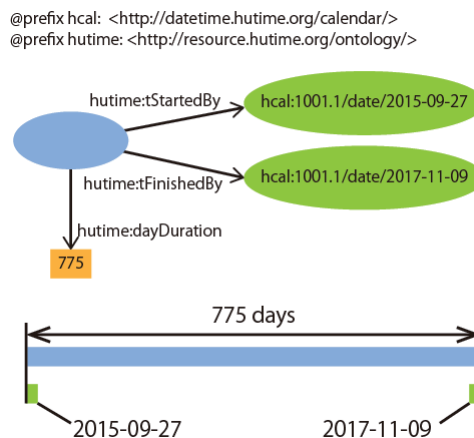


図 2 「2015年9月27日から2017年11月9日までの日数」を示す RDF。期間の端点として、起点の始点と対象点の終点が指定されており、RDF では Allen’s Time Interval Algebra で表現されている。下図は、RDF が示す期間の相対関係を時間軸に沿って示したもの。

Figure 2 RDF which shows day count from 27th September 2015 (origin) until 9th November 2017 (destination). Beginning of the origin and end of the destination is used to the calculation. These endpoints are shown by Allen’s Time Interval Algebra. The lower panel shows relative relationships between the time intervals along a temporal axis.

4. 利用例

以下に処理の例とそれを実行するための変数の設定例を示す。設定値内の改行 (CRLF) は“`␣`”で表す。暦 ID は“1001.1”と“101.1”が使われており、それぞれ、和暦（南朝）、および、西暦（ユリウス/グレゴリオ暦-1582年改暦）を示している。

(1) 暦日等に関する情報の取得

処理：暦年を表す文字列を、書式を定めて出力する。

```
method = info
ical = 1001.1
itype = year
ival = 延宝八年␣元禄15年␣宝永己丑
oform = gy年yE
```

結果：延宝8年庚申␣元禄15年壬午␣
宝永6年己丑

変数 method に処理の種類が設定され、変数 ical, itype, ival により、暦年が指定される。変数 ival に改行で区切られた3つの暦年を表す文字列が入力されており、これらは一括で処理される。出力される文字列の書式は変数 oform

に設定されており、バラバラな書式で表された3つの暦年が、算用数字の暦年と干支を組み合わせた書式に統一される。

(2) 暦法の変換

処理: 「元禄年間」に対応する西暦年を得る。

```
method = conv
ep = be
iuri = http://datetime.hutime.org/calendar/
      1001.1/era/元禄
ocal = 101.1 otype=yera
```

結果: C.E. 1688⇔C.E. 1704

暦日等は暦リソースのURIで指定されている(変数 iuri)。変換の基準になる端点は、始点と終点の両方が指定されているため(変数 ep)、それぞれに対応する2つの西暦年が改行で区切られて出力される。

(3) 暦日等への期間の加算

処理: 元禄14年3月14日の22か月後の暦日を得る。

```
method = addCal
ep = bb
ical = 1001.1
itype = date
ival = 元禄14年3月14日
ical2 = 1001.1
ival2 = +0-22-0
```

結果: 元禄15年12月14日

加算される期間は和暦に基づいており(変数 ical2)、22か月間は、年数、月数、日数の順で、“+0-22-0”と表現される(変数 ival2)。閏月(元禄15年閏8月)を含めた月数が計算される。

(4) 暦日等の間の期間の長さを算出

処理: 徳川綱吉の将軍在位期間を計算する。(延宝8年7月18日～宝永6年1月10日)

```
method = durCal
ep = be
ical = 1001.1
itype = date
ival = 延宝8年7月18日
ical2 = 1001.1
itype2 = date
ival2 = 宝永6年1月10日
```

結果: +28-5-23

出力される期間(年数+月数+日数)の表現は、期間の加算の例と同様であり、28年5か月23日

間を示す。在位期間を求めるため、端点は始点から終点が指定されている。

(5) 暦日等同士の間隔の取得

処理: 和暦の元禄年間とグレゴリオ暦の17世紀の間隔の取得。

```
method = comp
ical = 1001.1
itype = era
ival = 元禄
ical2 = 101.1
itype2 = century
ival2 = 17
```

結果: OverlappedBy

結果のOverlappedByは、元禄年間の開始から途中までが17世紀と時間的に重なることを示す(図1参照)。

5. 応答時間

処理速度の点でのHuTime Web APIの実用性を評価するため、応答時間を以下のとおり測定した。計測は、4章で示した5つの利用例をそれぞれ実際に実行することにより行った。各処理とも、1, 3, 10, 30, 100, 300, 1000件の暦日等の一括処理をそれぞれ100回行い、その応答時間(HuTime Web APIに対するHTTPリクエストから結果を受け取るまでの時間)の平均値を算出した。実運用中のサーバへの影響を避けるため、測定は同じ機能を持つクローンサーバを用いて行った。また、クローンサーバでの測定値の妥当性を検証するため、4章(2)の暦法変換を1000件実行する場合のみ、実運用中のサーバでも同様の方法で応答時間を計測した。なお、計測用のプログラムは、Web APIを提供しているサーバ上で動作するため(クローンサーバ、実運用サーバとも)、ネットワークの遅延については無視できる。

計測の結果、いずれの操作も件数の増加とともに応答時間が増えている(図3)。ただし、30件以下では件数や処理の種類による違いがほとんどなく、8ミリ秒未満であった。この範囲では、暦日等の処理そのものよりも、HTTPサーバ内の処理などにかかるオーバーヘッドが応答時間に影響していることがうかがえる。最も応答時間が短かったのは(5)の相対関係の取得(comp)で1000件を一括処理した場合の平均応答時間は19ミリ秒であった。一方、最も応答時間が長かったのは(2)の暦法変換(conv)で、1000件を一括処理した場合の平均応答時間は128ミリ秒であった。なお、同じ暦法変換の処理を実運用中のサーバで実行したときの平均応答時間は181ミリ秒であり、クローンサーバでの結果とかけ離れたものではなかった。

これらの結果から、1) 実運用中のサーバの性能評価にクローンサーバでの計測値を用いることは妥当であること、2) 実際に利用が想定される数十件程度の処理では、応答は悪くならないこと、さらに、3) 数十件程度での応答時間は、ネットワーク上での遅延時間に比べて十分に短いことが示された。結果として、実運用中の HuTime Web API は、十分に実用的な処理速度を有することが示唆された。

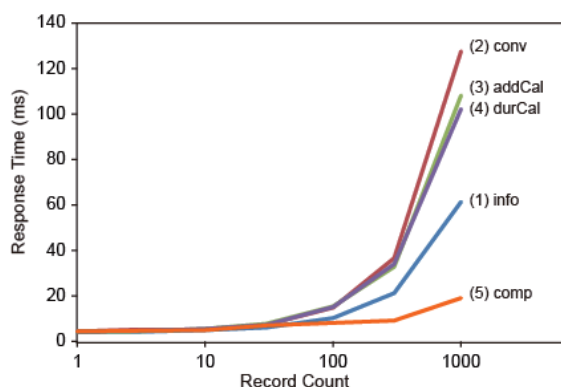


図 3 HuTime Web API の応答時間の測定結果. 図中の“info”から“comp”までの各項目は、テストで変数 method に設定した値で、それぞれ 4. 利用例の (1) から (5) までの各処理に対応する。

Figure 3 Measurement results of response time for HuTime Web API. Items from “info” to “comp” in the figure are value set on the “method” variable, and indicate processes from (1) to (5) in chapter 4, respectively.

6. まとめ

HuTime Web API では、暦日だけでなく年号、暦年、暦月など、暦法に基づいて表現されるさまざまな期間にも対応した。既存の多くのサービスは暦日のみが対象であるが、実際の検索や解析では、年号や暦年を単位として行うことも少なくない。それらの期間を直接扱えることは、HuTime Web API を使ったコンテンツやアプリケーションの利用者、開発者双方に大きな利便性をもたらすであろう。

また、暦法の変換だけでなく、期間に関する計算を実現したことは、本研究の最大の特徴である。さらに、これが Web API として提供され、外部から利用できることの意義は大きい。たとえば、事件やできごとの検索でこの機能を用いれば、「2年6か月以上継続した」や「元禄14年3月14日から22か月以内の」といった検索条件を設定するデータベースやアプリケーションを開発できる。しかも、これまでは困難であった和暦などの日付での直接指定が可能になるため、西暦に変換することによる手間や正確性の問題も克服できる。

また、セマンティック Web 技術や Linked Data と高い親和性を持つことも HuTime Web API の特

徴である。入力する暦日等の指定に Linked Open Data として公開している暦リソースの URI が利用でき、また、結果を RDF で受け取ることもできる。SPARQL のクエリから関数として呼び出す形での利用も検討されており、今後幅広い応用が期待される。

謝辞

本研究は、JSPS 科研費 15H01723 「セマンティック・クロノロジー：時間軸に沿った知識の可視化と利用に向けた基盤構築」の助成を受けたものです。また、部分的に、JSPS 科研費 16H01897, 16H01830, 16H02918, 17H00773 および 23300097 の助成を受けています。

参考文献

- 1) Suga, T.: hosi.org <<http://hosi.org/>> (参照 2017-11-15) .
- 2) 中央研究院資訊服務處：兩千年中西曆轉換 <<http://sinocal.sinica.edu.tw/>> (参照 2017-11-15) .
- 3) CalendarHome.com: Convert a date <<https://calendarhome.com/calculate/convert-a-date/>> (参照 2017-11-15) .
- 4) 関野 樹, 山田 太造: 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, 情報処理学会シンポジウムシリーズ, Vol. 2013, No. 4, pp.145-152 (2013) .
- 5) 関野 樹: 暦に関する Linked Data とその活用, 情報処理学会シンポジウムシリーズ, Vol. 2015, No. 2, pp.191-198 (2015) .
- 6) Sekino, T.: Basic linked data resource for temporal information. Proceedings of the 2017 Pacific Neighborhood Consortium Annual Conference and Joint Meetings (PNC), IEEE Catalog Number: CFP17M10-ART: 76-82 (2017).
- 7) Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM Vol. 26, No. 11, pp.832-843 (1983).
- 8) ISO: ISO 19108:2002 Geographic Information - Temporal Schema (2002).
- 9) ISO: ISO 24617-1:2012 Language resource management – semantic annotation framework (SemAF) – Part1: Time and events (SemAF-Time, ISO-TimeML) (2012).
- 10) W3C: Time Ontology in OWL (2017). <<https://www.w3.org/TR/owl-time/>> (参照 2017-11-15)
- 11) JSON-RPC Working Group: JSON-RPC 2.0 Specification (2010). <<http://www.jsonrpc.org/specification>> (参照 2017-11-15)