fhiso

CFPS 43
(Call for Papers Submission number 43)

Proposal to add style to the wholly-numeric representation of dates in CFPS 13

Submitted by: Smith, Richard

Created: 2013-04-19

URL: Most recent version: http://fhiso.org/files/cfp/cfps43.pdf
This version: http://fhiso.org/files/cfp/cfps43_v1-0.pdf

Description: Proposal to separate presentation from representation in calendars in order to avoid a proliferation of calendars.

Keywords: dates, calendars, date style, Julian calendar, Roman calendar, regnal years

Abstract

This paper supports the proposal in CFPS 13 to use wholly-numeric representations of dates to allow for natural sorting, and it is shown that a proliferation of calendars is harmful to this aim. Three use cases are considered that are applicable to mediæval European genealogy: years starting on Lady day, the Roman style of reckoning days, and regnal years. Each might seem to demand a separate calandar, but it is argued that the standard Julian or Gregorian calendar suffices. An appendix is used to show that a proposal for separate calendars, while retaining natural sorting, would result in contrived and counter-intuitive representations. Instead of a multiplicity of calendars, this paper proposes that the important details of the style of original date should be attached to date.

1 Naturally sorting calendars

The 'proposal to accommodate alternative world calendar system' found in CFPS 13 proposes an 'all-numeric representation of dates' [1] similar to and compatible with ISO 8601. According to the proposal, this results in an 'ordering of fields from coarse-grained to fine-grained, thus giving a natural sorting capability.' This paper defines a calendar as *naturally sorting* if a lexicographical sort of the representation is sufficient to place the dates in chronological order.

Natural sorting is particularly useful if an application is to behave gracefully with calendars it does not fully understand. So long as an application can sort dates correctly, it can index people chronologically, sort events by date, and order children from a marriage, all without understanding the calendar in use, so long as all dates are in the same calendar. Even when the application does understand the calendar, natural sorting is simpler to implement and therefore less error-prone. This paper therefore supports CFPS 13's implicit proposal that calendars are either recommended or required to be naturally sorting.

A *complete representation* of a date is defined in ISO 8601 as one that identifies a particular day, rather than a whole month or year [2]. For a complete representation of a date in the representation of the Gregorian calendar proposed in CFPS 17, lexicographical sorting is indeed sufficient to order dates correctly [3]. Although CFPS 17's representation of quarters are not naturally sorting, CFPS 40 proposes ways of resolving that so that all Gregorian dates are naturally sorting [4].

Whilst natural sorting makes it easy for applications to compare two dates in the same calendar, it does not provide a way of comparing dates in two calendars. In general, this can only be done if the application is able to convert the dates to be in a common calendar. This requires specialist knowledge of the calendar, can be complicated, and in the case of some ancient calendars not fully known.

This paper therefore discourages the proliferation of calendars insofar as possible. The following section examines three use cases which might seem to warrant separate calendars. (The appendix to this paper gives an example of the calendars that might result, were these use cases each to be realised in their own calendar while supporting natural sorting. The resultant calendars feel contrived.)

2 Use cases

2.1 Years starting on Lady Day

In much of mediæval Europe, it was usual to reckon years starting from 25 March, the Feast of the Annunciation, or Lady Day as it was commonly known [5]. This practice was abandoned in favour of the modern convention at various times between the 16th and 18th century, depending on the country in question. In England, 1752 was the first year to begin on 1 January. Thus, in most of Europe, the day after 24 Mar 1500 was 25 Mar 1501, and 25 Mar 1500 was 365 days earlier. In the historical literature, the abbreviation OS for 'old style' is sometimes appended to dates written using years starting on Lady Day; dual-style dates (such as 24 Mar 1500/01) are also be found.

2.2 Roman day reckoning

In the Roman calendar, days were counted backwards from three named points in each month — the Kalends, Nones and Ides [5]. This use continued in the mediæval period, and can be found in documents of interest to genealogists. A record from Barnwell Priory, for example, begins *septimo idus Julij* — "on the seventh [day before] the Ides of July". This should not be confused with *ad hoc* descriptions of dates such as 'the second Thursday after Easter'. The Roman calendar had a single, well-defined way of reckoning every day, with a standard abbreviated form such as *a.d. VII Id Jul* (where *a.d.* stands for *ante diem*, the Latin for "days before").

2.3 Regnal years

Another important class of date representation in use in mediæval and early-modern times is that of regnal years [6]. An example found in a Cambridgeshire churchwardens' accounts book is *iij^{o} die Mensis Maij anno r[egni] r[egis] henrici octaui sexto* — "the third day of May in the sixth year of the reign of King Henry VIII", which would commonly be abbreviated (in English) to *3 May 6 Henry VIII.*

(Regnal years are subject to several further difficulties. In England, King John's regnal years start from Ascension Day, a movable religious feast, which results in potentially ambiguous dates such as *5 May 3 John* — was this 1201 or 1202? In France, who was king between the death of Louis X and the birth of his only son, John I? How are restorations handled, such as of Henry VI in England in 1470?)

3 Separating calendars from style

The FHISO may reasonably take a view that these use case need not be supported; nevertheless, it seems unreasonable to say they *must* not be supported, and its likely that some applications would ignore such an edict if their users wanted support. But support for these use cases in data entry and in the display of dates does not require them to be directly supported in the serialisation format, and this paper proposes that they are not.

In the case of the Roman reckoning of days, it is reasonable to argue that *a.d. VII Id Jul* is just an alternative way of writing 9 July and it should be stored that way. (The appendix also shows that a direct representation of the Roman reckoning would be complicated and counter-intuitive if it is to be naturally sorting.) If the serialisation of dates writen in the Roman style is the same as those Julian calendar, is it useful to call them different calendars? The only remaining reason for wanting to call it a separate calendar is to record the fact that the date was originally specified with a Roman-style day, and there are better ways of recording what is essentially a stylistic variation.

Having established that *a.d. VII Id Jul* does not refer to a different calendar, it is natural to ask whether the same might be true of regnal years. Regnal years rarely begin on 1 Jan, and so the year, as written, changes midway through the year *anno domini*. This is similar to the situation with Roman day reckoning: *a.d. XIII Kal Jul* — thirteen days before the Kalends of July — is actually 19 June. The month included in the Latin abbreviation changes from June to July on 14 June, the day after the Ides of June. This is no different from the regnal year, where the displayed year changes from 5 Henry VIII to 6 Henry VIII on 22 Apr 1514. An application can still note that a date was styled using an English regnal year, while serialising the year *anno domini*.

Years starting on Lady Day are no different from regnal years that start at some point midway through the calendar year. The only real difference is that '1514' is a number, while '6 Henry VIII' is not; but just because 1514 is numeric does not mean it need necessarily be the same as the numeric year in the serialisation. Contemporary records referring to the execution of Charles I give its date as 30 Jan 1648, but it can still be serialised as `1649-01-30` in the Julian calendar. As in the other cases, the application may choose to note that the date was styled using years beginning on Lady Day.

4 Proposal

For the three specific use cases considered here, this paper proposes that they are *not* considered to be separate calendars. This is consistent with the earlier principle of discouraging the proliferation of calendars. All three use cases can be accommodated using the Gregorian calendar or a similar Julian calendar with years starting on 1 Jan.

The Gregorian and Julian calendars *do* qualify as distinct calendars because they have different rules as to the lengths of years and months. (If dates prior to the Augustan reform of the Julian calendar in *circa* 4 AD are to be handled, a further calendar and calendars would probably be needed; however such calendars are outside the scope of this paper, and the FHISO may feel they are outside the scope of what it should standardise.)

To avoid losing information about the original style of the date, this paper proposes that the FHISO specify a mechanism whereby multiple *calendar style IDs* can be stored alongside dates. This paper does not propose what this mechanism should be. It might be appending them to the date representation using some suitable markup, for example `#JU#1299-07-09@RO,RE_EN@` where `JU` is the calendar ID for the Julian calendar, `RO` is a style ID indicating that the source used the Roman reckoning of days, and `RE_EN` indicates English regnal years. Alternatively, an XML serialisation format might use attributes, and perhaps longer, more readable IDs.

```
<date calendar="Julian" style="roman regnal(England)"
 >1299-07-09</date>
```

This mechanism preserves the orthogonality between the calendar (Julian or Gregorian), and the formating of days (Roman or conventional), and the formating of years (regnal, starting from Lady Day, or conventional). If desired, the calendar style ID could also be used to encode the language the original date was written in, perhaps using ISO 639 language codes. The aim is not that the style encodes enough detail to precisely recreate the source text: rather that it should encode its important characteristics. As what is thought important will vary from genealogist to genealogist, all such style information would be optional.

Applications need not support any style IDs. Given three dates

```
<date style="lady-day">1649-01-30</date>
<date style="regnal(England)">1514-05-03</date>
<date style="roman regnal(England)">1299-07-09</date>
```

an application could simply render them all in the same style. And in some contexts, such as where many dates are listed together and it is important for user to be able to compare them visually, the application would probably want to do that even if it did support the specified styles. But in other contexts, such on a detailed event view page, an application familiar with each of these styles might perhaps display them as: *(i) 30 Jan 1648/49*, *(ii) 3 May 6 Henry VIII* (1514), and *(iii) a.d. VII Id Jul 18 Edward I* (9 Jul 1299).

This set of calendar style IDs, much like the set of calendar IDs, would need to be extensible — especially for regnal years, where it is unreasonable suppose the FHISO could standardise data for all relevant monarchies. One possible way forward, not explored in detail here, is to have some discovery mechanism by which an application can find out more about an unknown calendar style. If a calendars style ID has an associated URI (for example from resolving the namespace URI of a QName as is suggested in CFPS 37 [7]), perhaps the resource found at that URI could to yield information about the style.

The examples here refer to styles that, in the XML versions, are named `lady-day` and `roman`, plus a family of styles named `regnal` parameterised by a country name. These are just examples, which, if they are to be standardised, should be subject of future proposals. Similarly, a future paper may be required to specify the default style for specific calendars and to explain how proposals should manage the interactions between multiple styles, such as between `regnal(England)` and `roman` in the examples here.

Appendix

This appendix considers an alternative possibility: that separate calendars *are* used for years starting on Lady Day, for Roman day reckoning, for regnal years, and so on. This paper emphatically does not propose such a scheme, partly because of the unnecessary complexity that would result from it, and partly because of the undesirability to have so many additional calendars.

A.1 Years starting on Lady Day

To support the years starting on 25 March using the year as written, and without losing the ability easily sort dates, requires the representation to abandon the familiar numbering of months with `01` for January through to `12` for December. One possibility is to use `13`, `14`, `15` for January, February and the first half of March, and to say that the day before `1500-03-25` is `1500-15-24`.

A.2 Roman day reckoning

The usual representation of a Roman day is with a Latin abbreviation such as *a.d. VII Id. Jul.* The order can be flipped so that it conforms to ISO 8601's policy of putting larger time periods first, and months can have their standard numbers (modified, if necessary, to accommodate years beginning on Lady Day). It would be easy to assign the numbers `1`, `2` and `3` to the Kalends, Nones and Ides, respectively; and the abbreviated Latin *ante diem* can be dropped. But what of the numeric part? If the date is written `YYYY-07-3-07` then the following day, `YYYY-07-3-06` sorts before it. An alternative would be to replace the number n with, say, $100-n$, giving `YYYY-07-3-93`, but this seems contrived and confusing.

This is further complicated by the way leap days were handled. Traditionally the leap day was 24th February. In a leap year, 25 February was *a.d. VI Kal. Mar.* – the sixth day before the Kalends of March; yet 23 February was *a.d. VII Kal. Mar.*. The leap day was written *ante diem bis sextum Kalendas Martias* – the second sixth day before the Kalends of March – and abbreviated *a.d. bis VI Kal. Mar.* In the representation here, the 23rd and 25th would be written `YYYY-03-1-93` and `YYYY-03-1-94`, which leaves no room for the 24th. An extra component appears to be needed, just to handle the bissextile day, with `-1` appended to the representation of all normal days and `-0` to the bissextile day.

Before the Julian calendar reform of 46 BC, the situation was more complicated, with multiple intercalary days inserted into February.

A.3 Regnal years

In a regnal year such as *3 May 6 Henry VIII*, handling the day and year of the reign is straightforward, and the problems of the month are no different from those with years that begin on Lady Day. This suggests a natural representation of `KKK-06-05-03`, but what should the `KKK` be that denotes the monarch? One possibility is simply to assign numbers from some fixed point — in England this might be the Norman Conquest; in France, perhaps the accession of Hugh Capet. Three digits would be required to cope with states with many rulers, such as the Papal States. Henry VIII was the 20th English monarch since the Norman Conquest which gives an encoding of `020-06-05-03`. But the requirement for a start point seems arbitrary, and unlike day or month numbers, the monarch number is unfamiliar.

A.4 Combinations

To further compound matters, these styles can be used together. For example, a Roman day numbering might be used with years *anno domini* (which may start on 1 Jan or 25 Mar), or with regnal years. A date like *a.d. VII Id. Jul. 18 Ed. I* would need a calendar that *(i)* is Julian, *(ii)* uses Roman day numbering, and *(iii)* uses English regnal years. Such a calendar is of niche use, and it is easy to see that a combinatorial explosion of calendars would result from such a scheme.

The representation of this date would be `009-18-19-3-93-1`: `009` is the position of Edward I in the list of post-Conquest monarchs; `18` is the year of the reign; `19` represents July as 12+7 because the reign began in November; `3` represents the Ides; `93` means 7 days before the Ides (counting inclusively); and `1` means that the day is not the bissextile day. By contrast, the Julian represention proposed in the main part of the paper is `1299-07-09`. This seems simpler, more elegant and more readable, as well as avoiding a proliferation in calendars. And if an application happens not to support the calendar and has to resort to displaying the raw representation of the date, the latter is more likely to be meaningful.

References

[1] Tony Proctor, 2013, *Proposal to Accommodate Alternative World Calendar Systems* (CFPS 13), `http://fhiso.org/files/cfp/cfps13.pdf`

[2] International Organization for Standardization, 2004, *Data elements and interchange formats — Information interchange — Representation of dates and times* (ISO 8601:2004)

[3] Tony Proctor, 2013, *Proposal to Accommodate Gregorian Dates using a Modified ISO 8601* (CFPS 17), `http://fhiso.org/files/cfp/cfps17.pdf`

[4] Luther A Tychonievich, 2013, *Sortable, Versatile CFPS 17* (CFPS 40), `http://fhiso.org/files/cfp/cfps40.pdf`

[5] Claus Tøndering, 1996, *The Calendar FAQ*, `http://www.tondering.dk/claus/calendar.html`

[6] Sweet & Maxwell, 1962, *Guide to Law Reports and Statutes*

[7] Richard Smith, 2013, *Proposal for a scalable extensibility mechanism* (CFPS 37), `http://fhiso.org/files/cfp/cfps37.pdf`