fhiso

CFPS 68
(Call for Papers Submission number 68)

Comments on the serialisation of date ranges in CFPS 17 and CFPS 40

Submitted by: Smith, Richard

Type: A comment on a submitted paper

Comment on: 40

Created: 2013-04-30

URL: Most recent version: http://fhiso.org/files/cfp/cfps68.pdf
This version: http://fhiso.org/files/cfp/cfps68_v1-0.pdf

Description: Comments on alternative syntaxes for date ranges, and the need to define their semantics including the arithmetic of durations and dates.

Keywords: dates, date ranges, durations, precision, calendars

Abstract

This paper considers the syntax for dates proposed in CFPS 17, and its development in CFPS 40 into a system for expressing ranges of arbitrary duration. Certain syntactic differences from ISO 8601 are noted, and it is proposed that the syntax should be adapted to conform to the ISO standard. A vocabulary is introduced for distinguishing ranges from periods, as is currently done in GEDCOM, and for discussing the precision of dates. The precise semantics of a date range are considered, leading to the definition of a range as a semi-open range. Consideration is also given to the meaning and usefulness of negative duartions in ranges.

Not all durations permitted in ISO 8601 intervals have a well-defined meaning, and some such cases are considered, together with possible resolutions. The handling of durations in non-Gregorian calendars, including those not fully understood by the application, is also discussed. The paper concludes with a discussion of the alternative approach of specifying ranges with a beginning and end date, instead of using durations. Although this would avoid many complexities in the handling of ranges, it does not avoid the need for the FHISO to specify durations and their properties.

1 Syntax for duration-based intervals

It is common to encounter a date that is only partially known. For example, the GRO index of UK births, marriages and deaths only gives the quarter in which the registration took place. The ISO 8601 standard provides *reduced representations* of dates where only the year and month, year, or century are recorded [1] (although the proposal before the FHISO in CFPS 17 does not include century representations). Quarter representations are introduced in CFPS 17 and are represented by *yyyy*-Q*n* [2]. A quarter is deemed to begin on the first day of January, April, July or October.

An alternative syntax for quarters (*yyyy*-*mm*⊕3) is proposed in CFPS 40 that sorts correctly with respect to other dates in the year and was generalised to allow certain other forms of range [3]. This syntax is very similar to the *date*/*duration* form of interval introduced in ISO 8601. The following table translates the examples given in CFPS 40 into the syntax of ISO 8601.

CFPS 40	ISO 8601	alternative ISO 8601
1801⊕100	1801/P100Y	
1950⊕10	1950/P10Y	
1956-07⊕3	1956-07/P3M	
2013-04-14⊕7	2013-04-14/P7D	2013-04-14/P1W
1971-06-30⊕366	1971-06-30/P366D	1971-06-30/P1Y

This paper suggests that, *if* such a format is desirable, then a subset of the ISO 8601 syntax is preferable to the CFPS 40 because

— ISO 8601 is already standardised with many implementations;
— the `D`, `W`, `M` and `Y` suffixes reduce the potential for ambiguity in the duration being specified; and
— *if* the alternatives in the last column are accepted, the ISO 8601 syntax allows for more natural representation in some sitatuions.

Excluding time components, the syntax of an ISO 8601 duration is `P`$[y$`Y`$][m$`M`$][d$`D`$]$ or `P`w`W`: it does not permit years, months and days to be combined with weeks. Weeks are not permitted in XML Schema's `duration` type, nor does it have a separate week-duration type [4].

2 Ranges and periods

For all its perceived faults, GEDCOM distinguishes between a period and a range [5]. In GEDCOM's terminology, a *period* is a state the lasted for a prolonged period of time. A marriage, for example, has an associated period, lasting from an initiating event (the wedding) to a concluding event (an annulment or divorce, or the death of a participant). In GEDCOM, a period is written "`FROM` *date* `TO` *date*". On the other hand, a *range* is used for an event that would have occurred on a particular date, but precise date is unknown. All that is known are some bounds. In GEDCOM, a range in which both bounds are known is written "`BET` *date* `AND` *date*", with bounds treated inclusively. A reduced representation such as *yyyy* is defined in GEDCOM to a shorthand for the range `BET` *yyyy*`-01-01 AND` *yyyy*`-12-31`.

By contrast, ISO 8601 just has one such concept: the *interval*, which 'comprises all instants between the two limiting instants'. One representation of an interval is "*date*/*date*". It is not specified whether ISO 8601's interval is equivalent to GEDCOM's period or its range, and presumably it gets used for either.

In a genealogical context, periods are a higher-level construct than ranges. A period might plausibly be bounded by a two ranges when there is uncertainty over the end points of the period. Because of this, the same representation should not be used for both ranges and periods. This paper proposes that the reduced representations of CFPS 17, the duration-based intervals of CFPS 40, and the ISO 8601-compatible syntax in §1 are all interpreted as ranges, not as periods. Periods are not further discussed in this paper.

3 Precision

Another question ISO 8601 leaves unanswered is what meaning to assign to an interval like `1950/P18M`. It is allowed by ISO 8601's grammar and means an 18 month interval starting in 1950, so it must end in 1951 or the first half of 1952. However, the explicit 18 month uncertainty is misleading because there is an additional 12 months uncertainty in the date. It could equivalently and more descriptively be written `1950-01/P30M`. This paper defines the *unit precision* of a date or duration as the smallest unit present in the date or duration. (The term *accuracy* is used in ISO 8601, but this is misleading in a genealogical context where both accuracy and precision are relevant.) This paper proposes that in an interval formed from a date and a duration, the duration must have an equal or less unit precision than the date. Thus `1950/P18M` would not be allowed, but `1950-01/P30M` would be.

This is compatible with, and does not preclude the adoption of, the stricter requirement in CFPS 40 that the unit precisions of the date and duration must be same. However, the use of durations with less unit precision seem beneficial in one imporant use case: that of estimating a date of birth from the person's age on a particular date. As an example, in the 1871 census of England (taken on 2 April 1871), a man said he was 26. Assuming the information recorded was correct, his date of birth was `1844-04-03/P1Y`.

4 Semi-open intervals and negative durations

Adding one year (`P1Y`) to `1844-04-03` results in `1845-04-03`. If these two dates represent a range treated as an *closed interval* — an interval containing both end points — then the interval is actually one year and one day long. An attempted resolution might be to say that `1844-04-03` actually represents an specific (though unspecified) instant of time on that day, and that `1845-04-03` represents the same specific instant of time. The interval is then only one year, but not usefully so, as an additional day's uncertainty comes from the unit precision, and the interval is effectively a year and a day.

This paper proposes the first date is inclued in the interval, but that the second date is not. This is the *semi-open interval* used in the definition of ranges in many computer languages. This means that a duration of `P1D` is redundant when specified on a complete representation (that is, a full *yyyy-mm-dd*), and similarly for durations of `P1M` and `P1Y` on the relevant reduced representations.

The ISO 8601 specification of durations does not permit negative durations; however the XML Schema duration type (and the two subtypes used in XPath) does permit negative durations, with the '`-`' sign preceding the initial '`P`'. Negative durations lead to a clearer representation in the example of the man aged 26 on 2

April 1871: the date of birth could be written `1845-04-02/-P1Y`. This is clearer because the date in the representation is `04-02`, the date of the census, instead of `04-03`, as required with a positive duration; and the year, 1845, is just 1871 − 26.

5 Adding dates and durations

How should an interval like `2013-04-23/P1M10D` be interpreted? Because of the variable number of days in a month, it matters whether the days are added before or after the months. Ten days after `2013-04-23` is `2013-05-03`, and one month after that is `2013-06-03`. But one month after `2013-04-23` is `2013-05-23`, and ten days after that is `2013-06-02`. Which is correct? Unfortunately, ISO 8601 is silent on the issue.

In XPath 2.0, the addition of `dates` and `durations` is not defined. Instead it uses two subtypes of duration — `yearMonthDuration` containing only a years and months, and `dayTimeDuration` containing just days and time components — and defines separately how these are added to dates [6]. (In principle there seems no reason why a number of weeks cannot be included in the day-type components, as the numbers of days in a week is constant, but neither ISO 8601 nor the XML Schema allow that.) If the FHISO is to support ranges identified by a start date and a duration, it must define how to calculate the end date, and this paper suggests using XPath's approach of using years and months, or days, but not both. (The stricter requirement in CFPS 40 is compatible with this.)

Another difficulty arising from the variable number of days in a month is that naïvely adding a number of years or months to a valid date might result in an invalid date. What, for example, is one month after 31 October 2012 or one year after 29 February 2012? Again, ISO 8601 does not provide an answer, and there isn't a unique, obviously-correct answer. In CFPS 40, the problem is neatly avoided by only allowing months or years to be added to reduced representations. According to the XML Schema specification, the answers are 30 November 2012 and 28 February 2013, respectively. This approach has the significant advantage of consistency: that *yyyy-mm-dd* + `P1M` always results in a day in the month *yyyy-mm* + `P1M`.

However, technical considersations are not the only relevant ones. For the purposes of UK election law, for example, 18 years after 29 February is defined to be 1 March. No doubt other legal jurisdictions have reached different definitions, and records that use month ranges may not respect the local legal precedent anyway.

If the FHISO wishes to support ranges by specifying a date and duration, and *if* the year and month durations are to be supported with exact dates, then this paper suggests that the most sensible interpretation is the widest meaningful one. Whereas XML Schema always rounds invalid dates down, this paper pro-

poses that for the purpose of defining an range, an invalid dates produced by adding a duration to a specified date should always be rounded away from the specified date. For example, some sources may consider `2012-10-31/P1M` to include `2012-11-30` and others (such as those calculating the end point using XML Schema's rules) may not. As a range is an expression of the uncertainty surrounding a date, it is range must include the possibility that `2012-11-30` is included. Similarly, `2012-10-31/-P1M` must include `2012-10-01`.

6 Working with arbitrary calendars

Although the discussion in CFPS 17, CFPS 40, and the preceding sections of this paper has been specific to Gregorian calendar, the same considerations will apply to a greater or lesser extent in other world calendars. By having naturally sorting calendars, a calendar can compare dates without understanding the calendar. The same is not true for arithmetic on dates and durations, but in some cases, certain operations can be done without full understanding of the calendar.

The main reason anticipated for needing to calculate the end date of the interval such as `1845-04-02/-P1Y` is to determine whether two dates are *compatible*. Is that computed date of birth compatible with another source that gives `1845-04-01/P1M`? (It is, just.) This is an example of test that can sometimes be done without detailed knowledge of the calendar. If the ranges are semi-open, equality with the computed end-point of the range means incompatibility. If the rounding of invalid dates is done away from the specified end-point, then with a naturally-sorting calendar, the rounding is unnecessary. A lexicographical comparison with the invalid date representation will behave exactly the same as a comparison with the properly rounded date.

This means that an application does not need to know how many days there are in each month in order to handle durations of months and years. Similarly, an application does not need to know about the number of months in a year to handle year durations. This is of particular use in allowing applications to cope with obscure world calendar.

A scheme was proposed in CFPS 67 for associating a set of default calendar facets with a calendar [7]. In that proposal they were used to govern certain aspects of the formatting of dates. This section discusses reusing them to allow certain arithmetic operations to be done on dates in a calendar-neutral manner. One facet proposed was the `year-and-recurring` facet, used to indicate that the first component of the date representation is a year. If a calendar has this amongst its default facet, then an application can handle year durations, without further knowledge of the calendar.

Suppose an application does not know about the French revolutionary calendar, and encounters the date `14-04-10/P2Y`. If the application can determine the calendar has the `year-and-recurring` facet amongst its defaults, then it knows that `14` is the year, and therefore adding `P2Y` gives `16-04-10`. Another date, d, is compatible with that source if `14-04-10` $\leq d <$ `16-04-10`.

Similarly, the `day-in-month` facet (in conjuction with `year-and-recurring`) tells the application that the second component is a month. To handle months, an application must know how many months in a year. The `western-month` facet states that there are twelve, and so an application seeing these three facets together can handle arbitray durations of years and months. The Swedish calendar of 1700-1712 is an example of an obscure calendar that would probably have three facets. (It only differs from the Julian calendar by the rule determining the number of days in February.) However, an application unfamiliar with this calendar could still process year and month durations, assuming it could determine the default facets.

7 Range-based intervals

Many of the difficulties discussed in this paper disappear if a range-based interval syntax is adopted. Such a syntax is defined in ISO 8601, and is simply written with the first and last dates in the range, separated by a '`/`'. Intervals written in such a format are naturally sorting (by their first date), and the need to do arithmetic with them is much less. As noted above, the main reason anticipated for adding dates and durations is to determine whether two dates are compatible. But with a range-based syntax is used, no arithmetic is needed, and the test is calendar-neutral without needing to introduce calendar facets.

This paper does not express a view as to whether the FHISO should adopt duration- or range-based intervals, or allow both (as ISO 8601 does). The duration-based syntax has the advantage that it is sometimes closer to the original source. But in more complicated situations, such as where a date range is synthesised from the intersection of two separate ranges in different sources, the range-based syntax is clearer.

Irrespective of whether duration-based intervals are allowed, the FHISO will need to standarise a syntax for durations which is needed to record ages, much as GEDCOM's in `AGE` tag. (The duration syntax in GEDCOM is essentially the same as the non-ISO 8601 duration microform introduced in HTML 5 [8].) Applications will also benefit from being able to add and subtract durations and dates, for example to help users in calculating a date of birth from an age on a given date. The problem of durations and adding them to dates therefore still remains, even if they are not used in the representation of ranges.

It could then be argued that how to add or subtract dates and durations is an implementation detail of the application, and as such not in need of standardisation by the FHISO. Were the FHISO to support just one or two calendars (for example, just the Julian and Gregorian calendars), this argument might have some validity. But if it is anticipated that some applications will support a wider range of calendars, and may be confronted with dates written in unknown calendars, applications will need to know what assumptions they may or may not make about dates and durations.

References

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

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

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

[4] World Wide Web Consortium, 2004, *XML Schema Part 2: Datatypes (Second Edition)*, `http://www.w3.org/TR/xmlschema-2/`

[5] Church of Jesus Christ of Latter-day Saints, 1996, *The GEDCOM Standard (Release 5.5)*, `https://devnet.familysearch.org/docs/gedcom/gedcom55.pdf`

[6] World Wide Web Consortium, 2010, *XQuery 1.0 and XPath 2.0 Functions and Operators (Second Edition)*, `http://www.w3.org/TR/xpath-functions/`

[7] Richard Smith, 2013, *Proposal to extend the calendar style mechanism of CFPS 43 into an abstract formatting model* (CFPS 67), `http://fhiso.org/files/cfp/cfps67.pdf`

[8] World Wide Web Consortium, 2012, *HTML 5: A vocabulary and associated APIs for HTML and XHTML*, `http://www.w3.org/TR/html5/`