Fuzzy dates in military history databases

FZY-D, a proposed variation on the ISO 8601:2004(E) datetime format

February 2010 version 1.1

MGT-WP-01

MGT-WP-01: Fuzzy dates in military history databases – FZY-D: a proposed variation on the ISO 8601:2004(E) datetime format

MGT-WP-01: Fuzzy dates in military history databases – FZY-D: a proposed variation on the ISO 8601:2004(E) datetime format.

Notes

This document was created in Adobe InDesign. The fonts are Gill Sans and Courier New, set at 10pt/14pt.
Margins have been set to make this document printable on both A4 and US Letter paper.

Version History

v1.0 July 2009. Original text, not released
v1.1 February 2010. Revised for clarity, with some additional material.

MacGregorTech
3 Glendower Place
South Kensington
London SW7 3DU
United Kingdom

Website: www.macgregortech.com
Email: roy@macgregortech.com
Telephone: +44 (0)20-7584 7891

Contents

Fuzzy dates in military history databases

FZY-D: a proposed variation on the ISO 8601:2004 datetime format

PART ONE – THE PROBLEM

1. Introduction

The author is a database developer, technical author and amateur military historian with about 30 years of experience in the IT business and a particular interest in managing data related to World War Two British Special Forces. One of the problems encountered with recording and analysing military history data is that the dates associated with people, events, documents and images etc are often imprecise ("the precision problem"), with a wide variation in the nature and extent of that imprecision, sometimes also covering intervals[1] rather than specific points in time. This is further complicated by variation in the levels of confidence in the accuracy of those dates ("the confidence problem"). Together these issues create a 'date fuzziness' that does not seem to be addressed by current date and time data formatting standards: in fact this appears to be considered as such an inconvenience to many in the IT world that they wish it would just go away rather than stand up and confront it properly. In the author's view this is a significant, short-sighted, technical and commercial mistake.

This White Paper, then, attempts to tease out, clarify and examine the many underlying aspects of these issues and proposes an adaptation of an existing standard to allow, paradoxically, a concise and precise way of storing, manipulating and displaying vague military history date data through a comprehensive practical and conceptual framework.

Although this is primarily designed to address the author's technical problems in his personal area of interest, some account has been taken of general historical contexts, and the main purpose of this paper is to draw out wider and deeper thinking, knowledge and expertise. There is no attempt here to prescribe how the proposed format should be implemented in any programming language or database system, nor how the format should be structured in XML (for instance), as these topics are better left to specialists in their fields.

Time and place are, of course, inextricably linked, so the reader should also note the related forthcoming White Paper: ***Fuzzy locations in military history databases*** *– FZY-L: a proposed variation on the ISO 6709:2008 geospatial location format* (MGT-WP-02), which will discuss a similar adaptation, for similar reasons.

To start with, two examples may help to illustrate the precision and confidence problems : -

A – The Precision Problem. A published memoir briefly describes a minor World War Two skirmish behind the German lines in the North African desert, during which a soldier is injured and leaves the unit to which he is attached. The unit's official War Diary for that month is missing from The National Archives though, possibly lost in the unit's later move from Tunisia to Italy. There is no doubt about in which month the incident happened and therefore roughly when the soldier left, but the actual days are not known, pending further research, so both dates could be recorded as '*definitely sometime during* February 1943' or '*definitely sometime between (and including)* 1 February 1943 and 28 February 1943' or perhaps '*almost certainly* 12th February 1943, *give or take 7 days*'.

1 The word "interval" is used throughout instead of "period" to indicate a date or time range, as this conforms with their very specific meanings within the ISO 8601 specification.

The confidence level is high but the dates are imprecise, and the preference would be to represent each of these two single events (skirmish and soldier leaving unit) in a database as separate, though imprecise, dates rather than as separate and precise intervals, which would give them a subtly different meaning. This is relevant to investigating the sequence and significance of other events nearby in time and space.

B – The Confidence Problem. A grainy scratched out-of-focus dog-eared black and white photograph shows five grinning soldiers dressed in skiing gear on a snowy wooded slope, but nothing is written on the back. The photo is loose in a box owned by the widow of a soldier who is not in the picture, so the unit is known. But who are his pals, and where and when was the picture taken? These questions are relevant to establishing a definitive list of the unit's personnel in the face of incomplete wartime records.

From the unit's War Diary it's known that two mountain training exercises were held; both in Italy, and roughly one year apart. Initially the photo could only be identified as 'Five unknown men in either Piedemonte d'Alife or Terminillo in Italy, *definitely sometime between (and including)* 19 March 1944 and 3 April 1945'. Later re-examination of the photo positively identifies two of the men and probably a third. From personnel records it's discovered that one of the identified men only joined the unit in July 1944, so the photo must be from the second exercise, in Terminillo, narrowing down (a) the interval to 8 March 1945 to 3 April 1945, and therefore (b) the possible identities of the other two men.

However, comparison with annotated photos in another collection shows that pictures taken around Terminillo are in sharper focus, with mountains rather than gentle hills in the background, so the conclusion is that it's from the first exercise after all, and the two unidentified men could not be X and Y as suggested by a colleague.

The photo could be dated therefore as '19 March 1944 *or later* and *probably (or almost certainly) not later than* 30 April 1944', or alternatively, '*probably during the period of* 19 March 1944 to 30 April 1944'. These dates are quite precise but there is not an absolute confidence about one or both of them, and there are various ways to describe that fuzziness as the photo goes through its many stages of fuzzy dating.

2. Date and time data types

How are these fuzzy dates and their qualifying remarks of precision and confidence to be codified and stored in a database in such a way as to be easily, meaningfully and consistently searched for, sorted on, displayed and shared with others? Date, time and datetime standard data types in database systems and spreadsheets are sometimes strictly numeric, stored as the number of seconds elapsed since an arbitrary base position such as 1 January 1904 or 1970. Although this allows great precision, efficient storage, fast processing, convenient date mathematics and strict sorting rules, it is not much use for recording the dates of birth of middle-aged army officers in World War Two (or earlier), and cannot handle dates that may only be known to within one month or worse. Any coded qualification of such a date, even if numeric, also has to be handled in a separate field rather than being incorporated directly as an integral part, to simplify searching and sorting. Other standard data types (as opposed to user-defined data types) may have an alphanumeric representation but often have very strict data entry rules that in fact only allow precise[2] content.

These kinds of problems led the author to investigate and experiment with alternatives, not as a datetime expert, but merely as an end-user seeking a solid rather than fudged solution. What follows is a discussion of the perceived issues and their implications, conclusions drawn from research and practice, and proposals that are designed to help data sharing beyond the author's private research area: in other words, version 2.0 of an earlier implementation that is not strong enough to support wider, public use.

2 *Where a "precise" date is one that includes year, month and day and a "precise" time includes hours, minutes and seconds.*

3. Meanings and definitions

In this paper "date" generally includes "time", so means "datetime" unless otherwise specified, to allow a less tedious narrative.

Fuzzy dates are not new in historical research (see footnote 9), but a general trawl of the internet suggests that no-one has systematically tackled the problems around date fuzziness, and also that there are some fuzzy ideas as to what constitutes a fuzzy date: a few technical forum discussions, purporting to be about fuzzy dates, have been found regarding how to calculate intervals that could result in displays like '*nearly 4 hours ago*' or '*last week*' or '*next Wednesday*'. These, it seems to the author, are not so much about fuzzy dates as about fuzzy descriptions of *very precise but relative* dates, which are not the same thing. This is, of course, a debatable point, but otherwise ignored here.

As no statements have been found as to what constitutes a precise date, let alone a fuzzy one – only general background assumptions that these are self-evident – here is an attempt at some working definitions : -

" A PRECISE DATE IS A POINT IN TIME WHERE ALL OF ITS RELEVANT COMPONENTS ARE KNOWN AND UNQUALIFIED "

The inclusion of 'relevant' gets us out of the trap of implying that a date without a time is therefore fuzzy and, similarly, a date that only has year and month components[3] may be perfectly accurate and precise in its context. This definition seems also to work for an interval : -

" A PRECISE INTERVAL COMPRISES TWO POINTS IN TIME WHERE ALL OF ITS RELEVANT COMPONENTS ARE KNOWN AND UNQUALIFIED "

A fuzzy date definition follows on from that then, fairly smoothly : -

" A FUZZY DATE IS A POINT IN TIME WHERE SOME OR ALL OF ITS RELEVANT COMPONENTS ARE KNOWN BUT QUALIFIED, OR UNKNOWN OR UNCERTAIN "

Again, intervals seem to be catered for. These definitions, though, are entirely separate from any textual description of a date (fuzzy or not), and fuzzy *relative* date descriptions, as noted above, could be derived from either precise or fuzzy dates, but we are looking here for unambiguous codification of inherently fuzzy dates. This definition of a fuzzy date underlies all of the thinking in the following sections.

4. Background

The author has specialised, for about twenty years, in developing cross-platform (Mac and PC), client-server, multi-user desktop applications using a Rapid Application Database Development System called 4D[4]. Trained originally as a Systems Analyst and Architect on mainframes, applications have been designed, developed, debugged or enhanced in a variety of business sectors (finance, medical, logistics, entertainment etc) for small, medium and large organizations including Apple Computer UK, Elsevier Science, the Performing Right Society, some large international merchant banks and one of Bill Gates' private companies. One the earliest adopters of desktop-publishing in the UK, the author has designed and written technical or historical papers, newsletters, press releases, manuals, user guides, release notes and obituaries.

3 *The word "component" is used specifically to conform with ISO 8601 terminology..*
4 *http://www.4d.com*

The 4D language, very similar to Pascal (the original language of the Mac OS), has a comprehensive command set that can be used to develop and deploy systems on Mac and PC, and compiled to machine-code for and from either platform. 4D has fully-integrated backend, frontend and web server facilities.

Privately the author is also a Founder and the Secretary of a war veterans' association, currently in need of its own website, and maintains a comprehensive database, written in 4D, to hold the veterans' personnel details as well as contact details for relatives and other interested parties. It was in the management of the personnel database and related historic event, image and document analysis that the problems of how to store and display fuzzy dates were first encountered and solutions devised. The original 4D-based format for fuzzy dates is slightly shorter than that proposed here, but its essential elements have been in practical use for several years.

A number of functions were written in 4D to convert data between conventional and fuzzy date formats as well as validate input and display the results, and the author is now looking to rewrite these and develop others in another language, specifically to redevelop and redeploy the database as a web application for fundraising purposes and to enable standard data interchange with other organizations. This is the driver for the paper: to fundamentally and comprehensively solve the fuzzy date problem in an open-source, web-standard way.

PART TWO - CORE FORMAT

5. ISO 8601:2004(E) and FZY-D

The most rational starting point for the formatting of fuzzy dates has to be the current International Standard for date and time representation and interchange, ISO 8601:2004(E)[5] (from here on simply referred to as ISO 8601), and this paper assumes that the reader is conversant with its details, so its contents are not repeated here except for the final paragraph of the Introduction : -

" *To avoid confusion between the representations and the actual text, its punctuation marks and associated graphic characters, all the representations are contained in brackets [ ]. The brackets are not part of the representation, and should be omitted when implementing the representations. All matter outside the brackets is normal text, and not part of the representation. In the associated examples, the brackets and typographical markings are omitted.* "

5.1 Existing ISO 8601 datetime format

Although ISO 8601 does appear to cater for fuzzy dates, in its definition of "*Representations with reduced accuracy*" *(section 4.1.2.3)*, in practice this 'representation-by-omission' leaves room for ambiguity of meaning. While a date can be fuzzy, its meaning should still be as precise as possible. For example[6] : -

`1985-04-12`	clearly this is 12th April 1985…
`1985-04`	…but does this mean 'April 1985' or 'a specific day in April 1985, not yet known'?

While a solution to this problem, like `1985-04-00`, might sort of work for a date, it doesn't deal with the difference between "*don't know, but wish I did*" and "*actually not needed, thank you*", and falls down immediately with a time, where `T12:30:00`, for instance, already has a specific meaning. What is proposed, to manage this and other problems, is a variation of, and extension to, the 20-character ISO 8601 extended date format (and to the 41-character ISO 8601 extended interval format), including time zone : -

`YYYY-MM-DDThh:mm:ssz`	ISO 8601 extended date format
`1985-04-12T12:34:06Z`	example

`YYYY-MM-DDThh:mm:ssz/YYYY-MM-DDThh:mm:ssz`	ISO 8601 extended interval format
`1985-04-12T12:34:00Z/1985-05-06T17:45:00Z`	example

Note that the extent of these base formats' level of precision goes as far as, but no further than, seconds, and this is deliberate. Although the author's experience of recording military history only extends to minutes (for example: "*The time-on-target bombardment started at precisely 0355 hours*"), the seconds time component is included for balance and completeness (compared with the date components) and because experience says that someone somewhere will have a use for it. Precision beyond that, into decimal parts of seconds, is not catered for at all in this paper, but is by no means inherently excluded.

5 *http://en.wikipedia.org/wiki/ISO_8601*
6 *The examples shown throughout are not necessarily displayed in the full ISO 8601 or FZY-D formats, for clarity.*

5.2 Proposed FZY-D fuzzy datetime format

The ISO 8601 standard explicitly allows for some variation, subject to "*…mutual agreement of the partners in information interchange." (section 3.7),* and the proposed new format is built very firmly upon those ISO 8601 foundations. Starting with 'the naming of the parts' it is proposed that this new format is identified by the name '**FZY-D**' (meaning Fuzzy Date, pronounced *fuzzy-dee*), with the suffix included to distinguish it from the complementary '**FZY-L**' (meaning Fuzzy Location, pronounced *fuzzy-ell* – see forthcoming White Paper MGT-WP-02). More formally it could possibly be called '**FZY-D (ISO 8601)**' to indicate its parentage without claiming that it's a ratified standard. Collectively, this 'fuzzy format family' could be called '**FZY-X**' (pronounced *fuzzy-ex)*.

Drawing on long experience and accepted good practice, it was decided that the ground rules for this FZY-D format were that it should aim to be : -

- precise – in the sense of precision of meaning
- explicit – if a component is not precisely known, this should not be just implied by its absence
- unambiguous – precision of meaning again
- comprehensive – to cover as wide a range of circumstances as reasonably possible
- adaptable – the format should allow for usage not envisaged by the author
- common-sense – use of symbols whose meaning is as clear, obvious or intuitive as possible
- human-readable – using extended format only, for ease of understanding the raw data
- sort-sensible – fuzzy and precise dates should group in rational ways
- fixed-length – to help manipulation with Regular Expressions
- ASCII friendly – all special symbols should be lower than decimal ASCII 128

It is proposed that the FZY-D format should extend the 20-character ISO 8601 base format by a further 9 characters, so that a 29-character FZY-D Datetime comprises the following parts : -

FZY-D Date – 10 characters as in an extended format ISO 8601 date, including hyphen separators
FZY-D Time – 9 characters as in an extended format ISO 8601 time, including 'T' prefix and colon separators
FZY-D Zone – 1 character as per the ISO 8601 Time Zone Designator, but with extra symbol options
FZY-D Era – 3 characters, plus prefix hyphen separator (represented as 'eee')
FZY-D Confidence Qualifier – 2 characters, plus prefix hyphen separator (represented as 'cc')
FZY-D Orientation Qualifier – 2 characters (represented as 'oo')
…including…
FZY-D Precision Indicators – substitute characters in various places to show imprecision

`YYYY-MM-DDThh:mm:ssz-eee-ccoo`	FZY-D Date format
`1985-04-12T12:34:00Z-CE#-?#<<`	example

A FZY-D Interval is simply the concatenation of two FZY-D Datetimes to give a 59-character string, including the standard ISO 8601 solidus [/] separator (but see PART THREE – FORMAT EXTRAS) : -

`YYYY-MM-DDThh:mm:ssz-eee-ccoo/YYYY-MM-DDThh:mm:ssz-eee-ccoo`	FZY-D Interval format
`1985-04-12T12:34:06J-AD#-**##/1985-05-06T17:45:00J-AD#-..##`	example

The fixed-length requirement follows naturally from other ground rules: leaving out any component would fatally conflict with unambiguity and human-readability, as well as with searching and sorting, and storage cost-per-

byte ceased to be an issue a long time ago. Similarly, the space character (decimal ASCII 32) would not be allowed anywhere in the FZY-D format, because it would also break one or more of the ground rules. All alphabetic characters would be uppercase only.

It is recognised that 'FZY-D Date' and 'FZY-D Datetime' would probably come to be used interchangeably, diluting the narrower definition of 'FZY-D Date' above – such is life – but would then usefully contrast with the term 'FZY-D Interval'.

These parts are explained and examined in detail in the sections that follow, but not necessarily in the order in which they appear above.

6. FZY-D Precision Indicators

FZY-D Precision Indicators are the substitute characters for the FZY-D Date, FZY-D Time and FZY-D Zone parts that make it possible to explicitly show which components of a FZY-D Datetime are imprecise. They are also used, identically or similarly, in the FZY-D Era and FZY-D Confidence and Orientation Qualifier parts.

Only one or the other of two characters are needed to indicate precision and get away from 'representation-by-omission' in ISO 8601 : -

Chars	*Decimal ASCII*	*Meaning*
#	35	Irrelevant, not applicable
?	63	Relevant but unknown

For example : -

`1985-04-##` April 1985, day not relevant
`1985-07-??` July 1985, but on what day?

Similarly, wider and wider dates could also be represented : -

`1985-##-##` 1985 – a year
`198#-##-##` the 1980s – a decade
`19##-##-##` 20th Century
`1###-##-##` 2nd Millennium

…and this is much neater than explicitly representing a year, for instance, as an interval : -

```
1985-01-01/1985-12-31
```

This also allows for some special variations (but see the comment about mixing, below) : -

`1985-1#-##` 1st Half of 1985
`1985-2#-##` 2nd Half of 1985

`1985-#2-##` 2nd Quarter of 1985
`1985-#3-##` 3rd Quarter of 1985

…and here are some further general examples : -

`1985-04-12T12:34:##`	12.34pm on 12 April 1985, seconds irrelevant
`1985-04-12T12:34:00`	this is also 12.34pm on 12 April 1985, of course
`1985-04-??T##:##:##`	unknown day in April 1985, time not relevant
`1985-04-12T00:00:00`	midnight at the start of 12th April 1985
`1985-04-12T??:??:??`	12 April 1985, time unknown but relevant down to the seconds

Either of the Precision Indicator characters could be used in any and all numeric positions, not excluding : -

`????-??-??T??:??:??`	date is completely unknown
`####-##-##T##:##:##`	date is completely irrelevant (someone will surely have a use even for this)

However, mixing or part-using FZY-D Precision Indicators within a component would not be allowed, with the exception of the special variations for Half-years and Quarters above (but see PART THREE - FORMAT EXTRAS) : -

`1985-04-?#T12:34:00`	not allowed: what would it mean?

A variation like `1985-04-2?` (meaning '*sometime between the 20th and the 29th*'), seems a bit grubby, so probably shouldn't be allowed, and is better handled in other ways, like using a FZY-D Interval (see also PART THREE - FORMAT EXTRAS). However, it would be acceptable for the FZY-D Precision Indicators to be non-hierarchic, meaning that lower order components could still be specified even if a higher order component is unknown, in an explicit departure from, and not possible under, ISO 8601 : -

`1985-??-12T12:34:00`	the year, day and time are known but not the month: why not?
`????-04-12T##:##:##`	only the month and day are known

Although a case could be made that a date component might be *relatively* unknown as opposed to *absolutely* unknown, therefore implying the adoption of one or more alternative 'unknownness' indicator characters, it's suggested that this nuance is better handled by the proposed FZY-D Confidence Qualifiers discussed below in Section 9.1, leaving clean and unambiguous meanings for both 'unknown' and 'irrelevant'.

7. FZY-D Zone Designators

It is proposed that a single 'FZY-D Zone' designator character is used, as in ISO 8601, in character position 20. The symbol set should comprise the full complement of all 25 NATO time zone designators, with 'Z' (that is, UTC) as the default[7], including : -

Chars	*Decimal ASCII*	*Meaning*
J	74	Local Time – this is the NATO designator for an observer's current local time

In addition though, either of the two FZY-D Precision Designators could be used : -

7 *www.timeanddate.com/library/abbreviations/timezones/military/*

`1985-04-12T12:34:##J`	12.34pm local time
`1985-04-12T12:34:##R`	12.34pm New York time (UTC -5 hours: Eastern Standard Time)
`1985-04-12T12:34:##?`	12.34pm, time zone unknown
`1985-04-12T12:34:###`	12.34pm, time zone irrelevant

Consideration was given to questions such as how to represent times like 'Double Summer Time' (as was the case in Britain during World War Two), or 'Pacific Daylight Time', possibly by using single lower-case characters or by adding more characters to the FZY-D Zone code, but nothing was concluded. These are probably best handled by computation from date and location data, and the 'Local Time' concept helps us out of that trap.

8. FZY-D Era Indicators

The Era part of the FZY-D format is the most obvious departure from the author's narrow historical view, but would seem to be quite logical to include, and differs from the use of the negative sign in ISO 8601 for years before year 0000 in the Common Era. ISO 8601 is of course based upon the Gregorian Calendar, and the four digits of the year component, along with the Era codes, limit us to a 20,000 year timespan (-2), but surely this is adequate for most of general as well as military history. In any case, ISO 8601 allows extensions to the year characters, by mutual agreement, and the intention in this paper is to get to a basic and baseline agreement on historical fuzzy date encoding.

It is proposed that one of four 3-character FZY-D Era codes is used, preceded by a hyphen separator, in character positions 21 to 24 as follows : -

Chars	*Decimal ASCII*	*Meaning*
`AD#`	41 + 44 + 35	Anno Domini
`BC#`	42 + 43 + 35	Before Christ
`BCE`	42 + 43 + 45	Before Common Era
`CE#`	43 + 45 + 35	Common Era
	…plus…	
`###`	35 + 35 + 35	Irrelevant, not applicable (but probably defaulting to AD/CE of course)
`???`	63 + 63 + 63	Relevant but unknown
	…derived from the FZY-D Precision Indicators.	

For example : -

`1985-04-12T12:34:##Z-CE#`	12.34pm GMT on 12 April 1985 in the Common Era

The 3-character format also allows ample room for further FZY-D Era codes.

9. FZY-D Qualifiers and the representation of uncertainty

Whereas *irrelevance* and *unknownness* are about levels of precision and can be represented as substitute characters within the FZY-D format (the FZY-D Precision Indicators), *uncertainty* is about levels of confidence. This qualification of a date, regardless of its precision, cannot be adequately managed by further substitute characters, so must be encoded separately, and it is proposed that uncertainty is handled as a 4-character code, preceded by a hyphen separator, in character positions 25 to 29. However, research and experimentation with concepts of date uncertainty show that they can be split into the two sub-categories of 'confidence' and 'orientation'.

9.1 FZY-D Confidence Qualifiers

Date confidence is exemplified by words and phrases such as 'probably', 'approximately' and 'to be confirmed', and the following eight 2-character codes are proposed for character positions 26 and 27 : -

Chars	*Decimal ASCII*	*Meaning*
`##`	35 + 35	Unqualified
`**`	42 + 42	Definitely
`*#`	42 + 35	Almost certainly
`..`	46 + 46	To be confirmed
`?#`	63 + 35	Probably
`??`	63 + 63	Maybe
`~#`	126 + 35	Approximately, circa, around
`~~`	126 + 126	Very roughly

…but note that here, [##] means 'Unqualified', not 'Irrelevant', and [??] means 'Maybe', not 'Unknown': further reasons why the format is fixed-length, for unambiguous positional meaning. Conveniently there is no clash between the rising values of the ASCII numbers and the falling levels of confidence, for sorting purposes.

In addition, perhaps there should also be a catch-all code to handle situations that are just too complex to convey with any of the other codes : -

Chars	*Decimal ASCII*	*Meaning*
`++`	43 + 43	Careful, see notes!

For example : -

`1985-04-12T12:34:##Z-AD#-~#`	approximately 12:34pm on 12 April 1985
`1985-04-12T##:##:##Z-AD#-?#`	12 April 1985 (probably)
`1985-04-??T##:##:##Z-CE#-..`	unknown day in April 1985 (to be confirmed)
`1985-04-12T##:##:##Z-AD#-##`	12 April 1985
`196#-##-##T##:##:###-###-~#`	Around the 1960s
`1960-##-##T##:##:###-###-~#`	Circa 1960
`1945-09-##T##:##:###-###-~~`	Very roughly September 1945
`1066-??-??T##:##:##?-???-++`	Hmm, needs more research

It's recognised that some combinations of FZY-D Precision Indicators and FZY-D Confidence Qualifiers (as in one example above, with [??] in the day position and [..] as the uncertainty qualifier) might need to be proscribed as redundant or too confusing to be meaningfully interpreted, but these are issues of implementation rather than format. There is also a subtle limitation here: it's not possible to indicate an approximate date with a definite time (or vice versa), or work on an even finer granularity – the level of individual date or time components – it is all or nothing, and probably doesn't much matter anyway (unless, of course, someone knows better).

9.2 FZY-D Orientation Qualifiers

The word 'orientation' was chosen deliberately, after considering and discarding 'breadth', 'width', 'range', 'scope' and 'direction' as not quite conveying the appropriate concept.

Date *orientation*, then, is exemplified by words and phrases such as 'before', 'or later' and 'week ending', and the following eleven 2-character codes are proposed for character positions 28 and 29 : -

Chars	*Decimal ASCII*	*Meaning*
<#	60 + 35	Week ending[8]
<<	60 + 60	Before
<=	60 + 61	Or earlier
<>	60 + 62	Mid
=<	61 + 60	Early
=>	61 + 62	Late
>#	62 + 35	Week beginning
>=	62 + 61	Or later
>>	62 + 62	After
	…plus…	
##	35 + 35	Irrelevant, not applicable
??	63 + 63	Relevant but unknown

…exactly as in the FZY-D Precision Indicators, so here [##] *does* mean 'Irrelevant', and [??] *does* mean 'Unknown'.

So, now we have the full 29-character FZY-D Date format. For example : -

`1985-04-12T12:34:##Z-AD#-##>=`	12:34pm on 12 April 1985 (or later)
`1985-04-12T##:##:##Z-AD#-?#<<`	Probably before 12 April 1985
`1985-04-12T##:##:##Z-AD#-~#<#`	Approximately the week ending 12 April 1985
`1985-04-##T##:##:##Z-CE#-##<>`	Mid April 1985
`19##-##-##T##:##:###-###-##=>`	Late 20th Century

The alert reader will notice that there may be an overlap of meaning or concept between some of the FZY-D Confidence and FZY-D Orientation Qualifiers (for example: 'Approximately' and 'Mid'), but it is felt that in practice these sub-categories are different enough to separate them like this, and that either way they provide a rich vocabulary and great flexibility to fuzzy date encoding and description, without significant loss of precision in meaning. How this information is stored and how it is displayed are separated, like HTML and CSS.

8 'Week ending' is relevant to recording data about WWII British Army Forms W3008 and W3009 "Field Returns of Other Ranks" and 'Field Returns of Officers'.

PART THREE – FORMAT EXTRAS

10. Other considerations

This part considers some exceptions to and variations on the core format proposals and examines issues that don't fit elsewhere. It should be noted though that foreign language matters are not considered at all.

10.1 Searching, sorting and date mathematics

Quite a lot of thought went into the selection of the FZY-D substitute, indicator and qualifier characters with regard to the implementation issues of searching and sorting. The FZY-D Era codes remain a problem though, and two other practical issues surfaced when the author was transcribing a unit's War Diary into a database using the original version of the FZY-D format : -

- should FZY-D Intervals sort before or after FZY-D Dates?
- should dates containing unknown components sort before or after precise dates?

For example, although one might think that a War Diary simply has only one slot per day, in practice the author has seen data something like this (format: dd.mm.yy) : -

```
3.5.44      this happened to Company A
4.5.44      that happened to Company A and B
5.5.44      unit HQ moved to location X
1st - 7th   Company C away on training exercise
```

…and is in favour of being able to maintain that sequence as found in the original data. Using the FZY-D format as proposed would, of course, sort the interval to the top : -

```
1944-05-01T##:##:##Z-AD#-####/1944-05-07T##:##:##Z-AD#-####
1944-05-03T##:##:##Z-AD#-####
1944-05-04T##:##:##Z-AD#-####
1944-05-05T##:##:##Z-AD#-####
```

Consideration was therefore given to allowing alternative characters, above or below the decimal ASCII range of 30 to 39 (the numbers 0 to 9), to the proposed 'irrelevant' [#] and 'unknown' [?] characters. However, this is probably not best handled as a format issue, not least because it's essentially a complicating fudge, and using another field to manage sequence is probably better for allowing users to define how to group fuzzy and non-fuzzy dates.

How can meaningful date range searching and date mathematics be undertaken with fuzzy dates? Again, these are implementation issues best handled elsewhere, using Regular Expressions, but it may be useful to note that the author devised some (tacky) methods that dynamically substituted 'start-of-month' or 'end-of-month' values (for example: `YYYY-MM-01` or `YYYY-MM-31`) into fuzzy dates at runtime to get a handle on these kinds of problems. Clearly this needs looking at to devise industrial-strength solutions.

10.2 Miscellaneous

The author's investigations unearthed a 20-year-old document about historical research[9] that contained an interesting variation on the representation of intervals. In trying to convey 'sometime between this date and that', one can just concatenate two qualified FZY-D Dates to make a qualified FZY-D Interval, but this document described a triple-date format which conveyed 'this particular date, give or take a specific amount of time', and comprised a 'Probable Date', followed by a 'Minimum Date' and a 'Maximum Date'.

Not wishing to introduce an extension to the FZY-D format, consideration was given to a possible way of achieving this effect within the proposed format and general spirit of ISO 8601, like this : -

```
1985-04-12T##:##:##Z-AD#-?###±####-##-14T##:##:###-###-~###
```

…meaning '*probably 12 April 1985, plus or minus approximately two weeks*', by using the [±] character (decimal ASCII 177) as an alternative but quite intuitive separator. Obviously this doesn't allow an asymmetric span of the central date, but otherwise seems to adequately manage the problem. Although the alternative separator does break one of the ground rules by being above decimal ASCII 127, this exception is probably justifiable. Note that this should not be confused with the matching ISO 8601 format placeholder character for either of the plus *or* minus characters: this is the actual character [±].

Consideration was also given to the concept of 'alternative date' so, extending thinking from the [±] exception, we could usefully have another : -

```
1985-04-12T##:##:##Z-AD#-####|1985-04-13T##:##:##Z-AD#-####
```

…meaning '*either the 12th or the 13th April 1985*', with the pipe [|] character (ASCII 124) again quite intuitive.

The related concept of 'alias date' falls here too: we could use the combination of pipe separator with uncertainty qualifiers to convey subtleties such as '*probably this date but maybe that one*', or '*if you see this date you should also look at that one*'.

To clarify: if these variations were implemented then there would be three legal interval separators, where the solidus [/] and the pipe [|] both indicate that the second part of the format is also a date, while the plus/minus character [±] indicates that the second part of the format contains a positive integer value in one or more of the date and time component positions. For input validation purposes the solidus and pipe formats would indicate that the second date part must be greater than the first (equal would of course be meaningless) for FZY-D Era codes AD and CE, and the reverse for FZY-D Era codes BC and BCE.

Section 6 above looked at some special format variations that allowed the representation of Half-years and Quarters. This could be extended further to handle the seasons as loose date-groups too, including solving the tricky problem of 'winter' crossing a year boundary at one end or the other : -

`1985-SP-##`	Spring 1985
`1985-SU-##`	Summer 1985
`1985-AU-##`	Autumn 1985 (or `-FL-` Fall)
`1985-W<-##`	Winter 1984/5
`1985-W>-##`	Winter 1985/6

9 *Historical Social Research Vol 14, 1989, No 3. 98-104 Managing Art History Fuzzy Dates.*

The concept of 'disputed', originally assigned [!!], was considered but dropped as it would have to be complementary, rather than alternative, to the confidence qualifiers, implying yet another extension to the format, and this was felt to be going too far, as well as also probably best handled in implementation. [!!] was then briefly recycled as meaning 'Definitely', with [!#] as 'Almost certainly', but these were changed because their ASCII values meant that they would sort before [##], meaning 'Unqualified', which really had to come first.

11. Conclusion

All this gives us a truly complete representation of a datetime in a way that is not possible in ISO 8601 alone: we have date, time, zone, era, confidence and orientation, all neatly bundled in 29 characters, catering for a very wide range of circumstances in a rich vocabulary and amenable to automated searching, sorting and filtering.

This all started, as these things do, as a small and casual solution to a minor irritation. Pondering the ramifications led the author inexorably into a much bigger arena, and it is believed that there is something useful here for a wide range of historians and database designers. The paper is not presented as a definitive fait-accompli, but put forward as a sturdy framework for robust examination, collaborative refinement and then consensus adoption.

R.M.Paterson
February 2010