Internationalizing date-time API consistent with the Earth, Moon, and leap seconds

Jim DeLaHunt • jdlh.com • IUC44 • 15 October 2020

Earth, Moon, and abolishing leap seconds

...the curious astronomy and politics of time()

Introduction

- Jim DeLaHunt
  - http://jdlh.com, ☎ +1-604-376-8953
  - Vancouver, Canada
  - Consultant in multilingual websites; software engineer
- Slides: http://go.jdlh.com/iuc44s5t1 (has links)

Questions about time and calendars

- How long is a day?
- How long is a year?
- When does a minute have 61 seconds?
- How does Posix time conflict with UTC?

How long is a day?

- Are days longer in summer, shorter in winter?

How long is a day?

- Always the same length: from sunup to sundown

How long is a day?

- Always the same length: from sunup to sundown
  - Oh, you want fixed-length units of time?

How long is a day?

- Always the same length: from sunup to sundown
  - Oh, you want fixed-length units of time?
- 24 hours x 60 minutes/hour x 60 seconds/minute = 86,400 seconds

How long is a day?

- Is this thing related?
- How long from one sunrise to the next?
  - 86,400 seconds?
  - Same every day?

“Sunrise on the New Jersey shore with seagull”
By John Robinette. CC-Attribution-ShareAlike

The map is not the territory

- "Day" originates from **observing** the Sun
- 1 day = 86,400 seconds every day is a **model**
  - Observed day is the prototype of the model
- A model often resembles its prototype, but is not the same as the prototype
  - Differences of model-prototype can be wacky

How is this related to l10n and i18n?

- Localisation: fit product to reality of new market
- Internationalisation: models which make l10n easier, cheaper
- Beware of being blinkered by your models
- Human reality is complex, weird
  - Writing systems, names, time

time() and struct tm

- Python, Unix, etc. ⇐ POSIX 1.2008 ⇐ C lib
  - ⇐ Pope Gregory, 500 years ago
- time_t time(time_t *tloc)
  - "time in seconds since Epoch" [00:00 1 Jan 1970 UTC]
- struct tm *gmtime(time_t *t) , localtime(...)
  - "convert time since Epoch... into a broken-down time, expressed as UTC"

time() and struct tm

tm struct in <time.h>			
int	tm_sec	Seconds	[0, 60]
int	tm_min	Minutes	[0, 59]
int	tm_hour	Hour	[0, 23]
int	tm_mday	Day of month	[1, 31]
int	tm_mon	Month of year	[0, 11]
int	tm_year	Years	since 1900
int	tm_wday	Day of week	[0, 6] Sunday = 0
int	tm_yday	Day of year	[0, 365]
int	tm_isdst	Daylight Savings	Flag

time() and struct tm

- 4.16 *Seconds Since the Epoch*
  - actual time of day vs seconds since Epoch: “unspecified”!
  - each… day shall be… “exactly 86400 seconds”
  - time_t *t* == *tm_sec* + *tm_min**60 + *tm_hour**3600 + *tm_yday**86400 + (*tm_year*-70)*31536000 + ((*tm_year*-69)/4)*86400 - ((*tm_year*-1)/100)*86400 + ((*tm_year*+299)/400)*86400

How long is a day?

- 1. November, 2020
- In California?
  –
- In Yukon?
  –

Map of North America including states and provinces, by Kaldari and Halava, cc-by-sa

How long is a day?

- 1. November, 2020
- In California?
  - 25 hours
  - “Fall back” to Standard time
- In Yukon?
  -

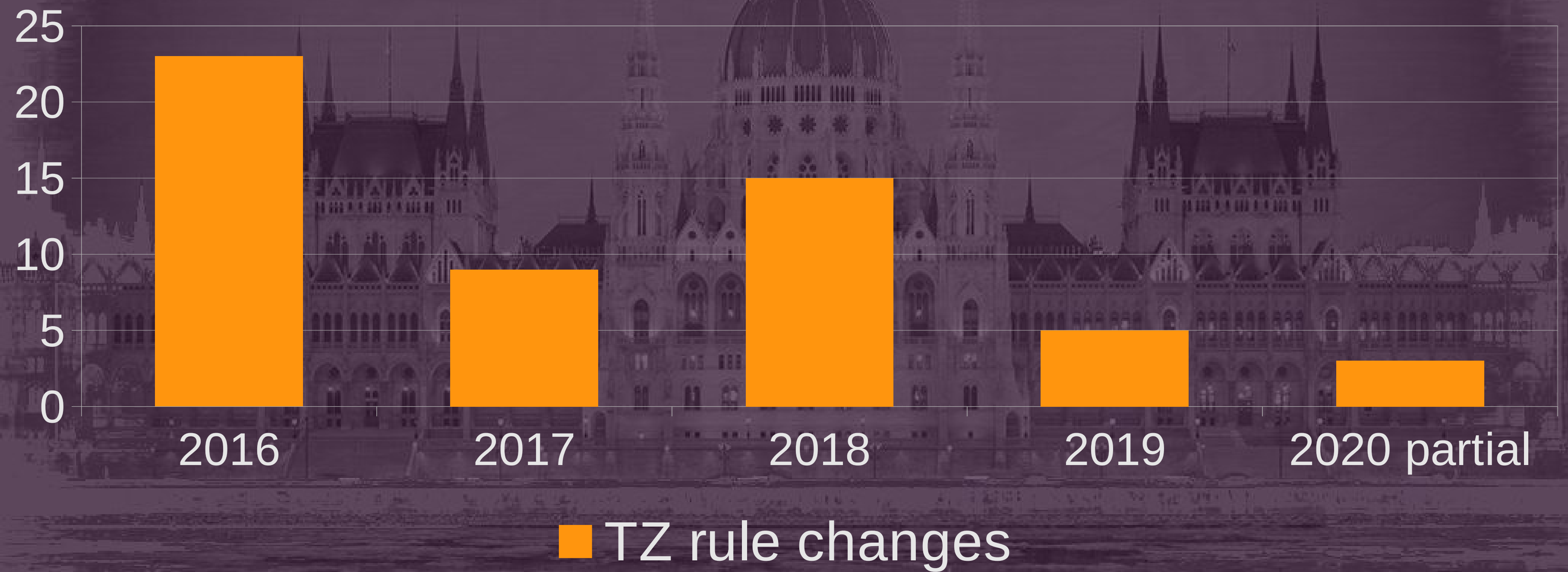
How long is a day?

- 1. November, 2020
- In California?
  - 25 hours, “Fall back” to PST
- In Yukon?
  - 24 hours
  - Not leaving daylight savings time
  - UTC-7 from now on

Time zone and Daylight Savings data

- Time zones and Daylight Savings time changes are human and political constructs
- They change
- IANA Time Zone Database
  - Was: Olson Database

Time zone rules are evanescent



Source: https://tzdata-meta.timtimeonline.com/

Seconds, days, and years

- Keeping track of seconds is timekeeping
- Keeping track of days and years is calendaring
  - “A calendar is a system of organizing days.”
- Two big things for a calendar to keep track of…

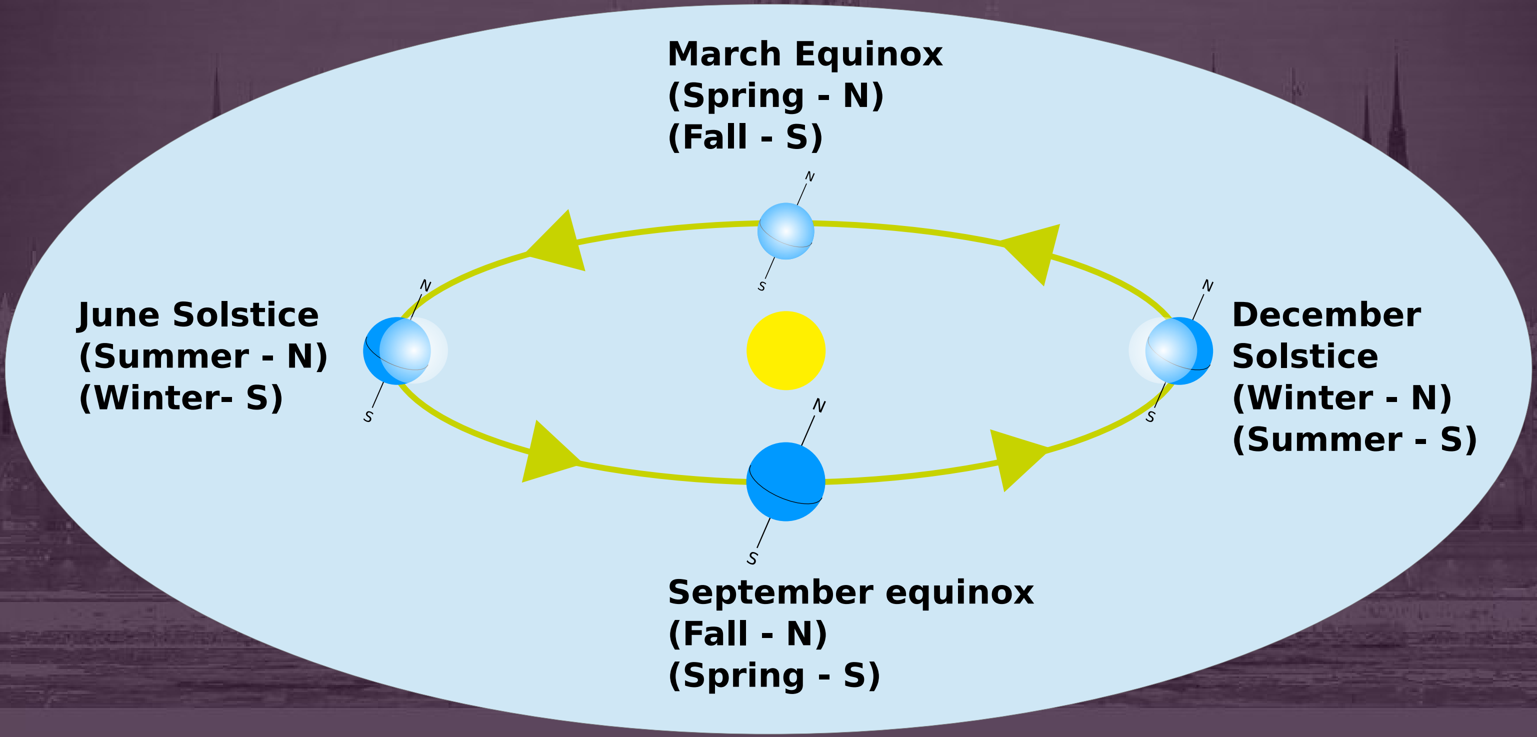
Two big things for a calendar to track

Sunrise in Florida, by Surge123. CC-BY 3.0.
Waxing half moon over Brofjorden, by W.carter. CC0.

Sun and moon

- Moon
  - “new moon”, between Earth and Sun, mostly dark
  - 29.5-day cycle approx
- Sun
  - March equinox, September equinox, solstices
  - 365.24-day cycle approx
  - Governs agriculture, so important

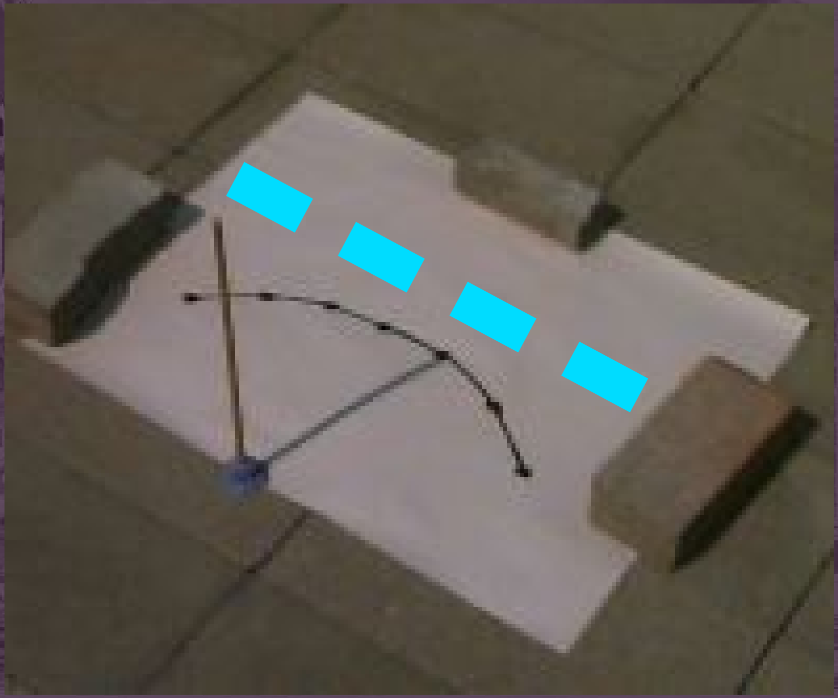
Sun, Earth, Equinox, Solstice



Orbital relations of the Solstice, Equinox & Intervening Seasons, by Colivine. CC0.

Indentify equinox with ancient tech

- Sundial, upright rod
- Plot shadow of rod tip
- On equinox, plot is a straight line



Fun with calendars

- Astronomy: 365.0-365.3 days/year
  - Tropical year, observed, March equinoxes
  - Synodic months, conjunction Moon-Sun
- Source: Urban & Seidelmann, *Explanatory Supplement to the Astronomical Almanac*
  - 3rd Ed. (uscibooks.com)
  - (archived summary)

Fun with calendars

- Julian: 365 (366) days/year.
  - Lost sync with solar year in year 45, 1580s
- Gregorian: 365 (366) days/year
  - Established 1582, to resync with solar year
  - 400-year cycle with 97 leap years

Fun with calendars

- Solar Hijri (“Iranian”): 365 (366) days/year
  - Starts on observed March equinox
  - Add leap day if equinox not yet arrived
  - (approx every 4-5 years, on 33-, 29-, 37-year cycles)
- Jalali: 365 (366) days/year
  - Predecessor to Solar Hijri, based on Hindi roots
  - Months on observed solar movement, 29-31 days

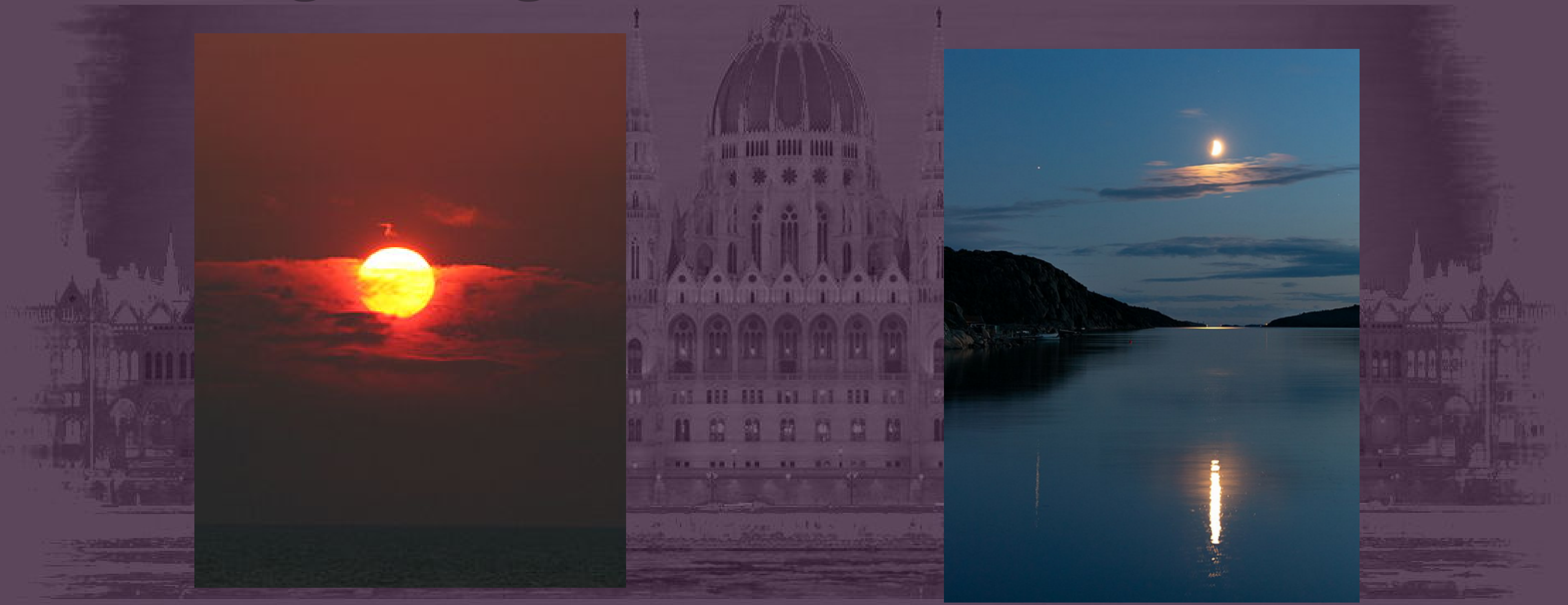
Fun with calendars

- Lunar hijri ("Islamic"): 354 (355) days/year
  - Purely lunar, month starts based on observation
  - Shifts through solar year
  - 30 year-cycle, 11 leap years
- Hebrew: 353, 354, 355 (383, 384, 385) days/year
  - Lunisolar, but calculated not observed
  - 19 year cycle, of which 7 have 13th month

Fun with calendars

- Chinese: 353, 354, 355 (383, 384, 385) days/year.
  - Timed by observation of moon and sun
  - 12 or 13 months per year.
  - 60 year cycle
- Indian: 365 (366) days/year
  - Since 1957, aligned with Gregorian calendar

Two big things for a calendar to track



Sunrise in Florida, by Surge123. CC-BY 3.0.
Waxing half moon over Brofjorden, by W.carter. CC0.

Time standards

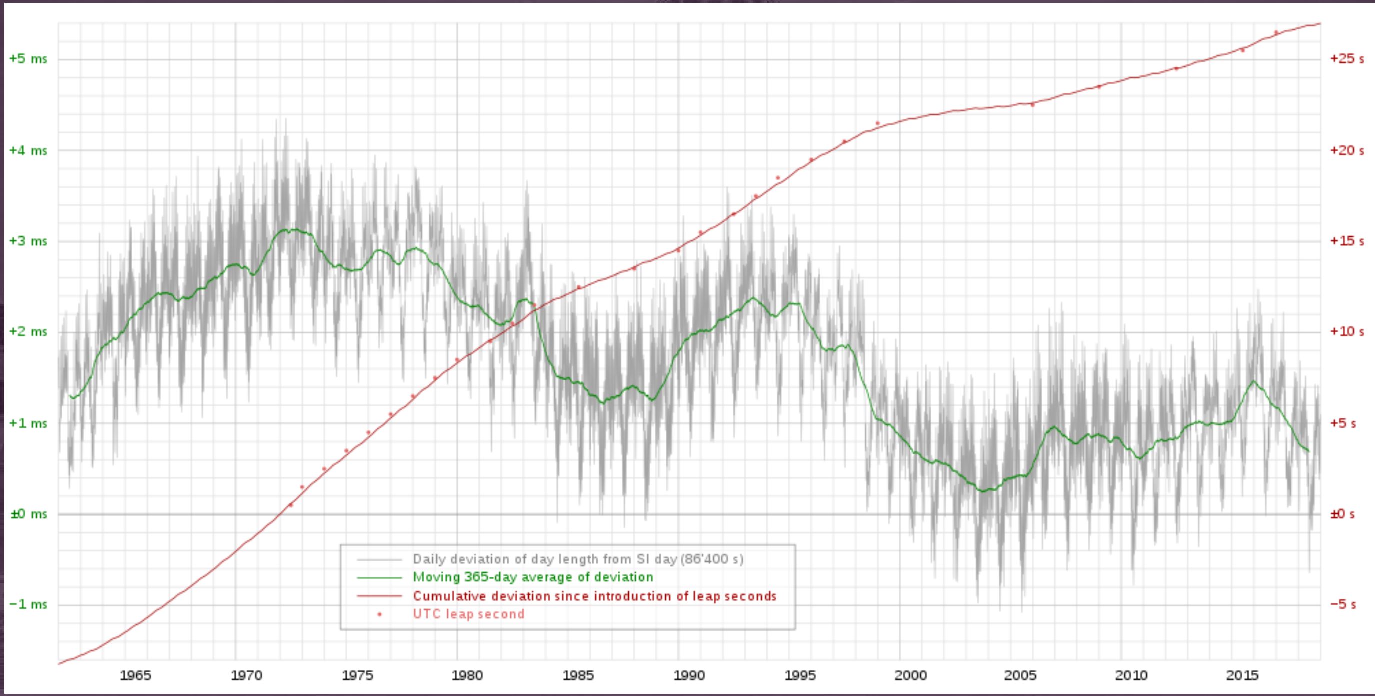
- TAI: pure linear count of SI seconds, atomic clocks
- UT1: universal time, mean solar time at 0° W.
  - UT1 days are not 86,400 SI seconds long
- GPS: TAI – 19 seconds

Deviation of day length from 86,400s

+5 ms

+25 s

0 ms



1970

2015

Time standards: UTC

- UTC: coordinated universal time
  - Same SI seconds as TAI (constant duration)
  - Within 1 seconds of UT1 (solar time)
  - Add leap seconds to keep UTC close to UT1
  - Integer number of SI seconds behind TAI (now: 37s)
- UTC shows: humans choose the reality of solar position over the perfect model which is TAI

Leap seconds

- Last minute of 30 June or 31 Dec UTC
  - As needed, per Int’l Earth Rotation… Service (IERS)
  - Makes the last minute have 61 seconds on some days
  - Clock reads “23:59:60”
- 27 leap seconds in 49 years (1972-2020)

Leap seconds and time()

- Recall, 4.16 *Seconds Since the Epoch*
  - actual time of day “unspecified”!
  - each… day shall be… “exactly 86400 seconds”
- POSIX (Std C) punts the leap second
  - To be fair, tm_sec does allow [0,60]
- Also, time() assumes drifting system clocks
  - Leap seconds are less severe than the drift

Leap seconds and time-critical tasks

- Some time-critical automated tasks
  - Computing, finance trading, automation
  - Increasing time accuracy: ms, ns
- Unpredictable leap seconds are a problem
  - No good system for advance notice of leap seconds
  - 23:59:60 UTC is during work hours (some timezones)

Proposal: abolish leap seconds

- Debated 2005-present, not resolved
- Case for abolishing
  - Problems for highly precise time-critical work
  - Poor POSIX support for leap seconds
  - Poor distribution of leap second advance notice
  - Having multiple timescales is confusing

Proposal: abolish leap seconds

- Case for retaining
  - Time-critical tasks, use TIA instead of UTC
  - Use better time sync than Network Time Protocol
  - Leap second << 1 hr daylight savings time change
- If no leap seconds, then calendar drifts out of sync with solar year

Meditation on time scales

- 3 goals, pick any 2
  - Precise time (or rather, stable precise frequency)
  - Simplicity (e.g. 86,400 seconds = 1 day)
  - Connection to solar days and solar years
- Advocates pick their goals, disregard the other
- Why not multiple time scales?
  - Adapted from *Plots of the time dilemma*, Steve Allen

Time to return to l10n and i18n

- Localisation: fit product to reality of new market
- Internationalisation: models which make l10n easier, cheaper
- Human reality is complex, weird
  - Writing systems, names, time
  - Also timekeeping and calendars

Time to return to l10n and i18n

- Humans can’t resist the solar day and year
- Precise, simple, solar-connection: pick any 2
- Beware of being blinkered by your models
  - Know when it’s time to create a new model
  - Be ready to advocate for reality over the model

Thank you!

- Q&A
- Slides: http://go.jdlh.com/iuc44s5t1

Internationalizing date-time API consistent with the Earth, Moon, and leap seconds

Jim DeLaHunt • jdlh.com • IUC44 • 15 October 2020