

tempuhs:

Arranging and presenting time

Alexander Berntsen & Stian Ellingsen, plaimi, 2014



tempuhs:

Arranging and presenting time

Where we are today

tempuhs:

Arranging and presenting time

Where we are today

→ Database

tempuhs:

Arranging and presenting time

→ Timespan json

parent	TimespanId	Maybe
clock	ClockId	
beginMin	ProperTime	
beginMax	ProperTime	
endMin	ProperTime	
endMax	ProperTime	
weight	Weight	
rubbish	UTCTime	Maybe

tempuhs:

Arranging and presenting time

- Timespans with timespans with timespans with timespans with...
- Records the relations between events
- Acknowledges uncertainty
- Preserves precision between systems
- Prioritises more relevant events

tempuhs:

Arranging and presenting time

→ TimespanAttribute json

timespan

TimespanId

name

Text

value

Text

UniqueTimespanAttribute

timespan name

tempuhs:

Arranging and presenting time

- Timespan attributes with timespan attributes with... only joking
- Simple key->value pairs

tempuhs:

Arranging and presenting time

→ **Clock json**

name

Text

UniqueClock

name

tempuhs:

Arranging and presenting time

→ Each event in time is recorded with a clock

tempuhs:

Arranging and presenting time

→ Role json

name	Text
UniqueRole	name

tempuhs:

Arranging and presenting time

→ Permissions

timespan

TimespanId

role

RoleId

own

Bool

read

Bool

write

Bool

share

Bool

UniquePermissions

timespan role

tempuhs:

Arranging and presenting time

→ Literally does nothing

tempuhs:

Arranging and presenting time

Where we are today

→ Server

tempuhs:

Arranging and presenting time

- Timespans, attributes & clocks can be posted to the server, and retrieved from it again
- Timespans may furthermore be marked as rubbish when obsolete
- Timespans may be filtered based on attributes

tempuhs:

Arranging and presenting time

Where we'd like to be tomorrow

tempuhs:

Arranging and presenting time

Where we'd like to be tomorrow

→ Universal magical unicorn time format

tempuhs:

Arranging and presenting time

- Problem: There is no universal clock
- Problem: We need a universal clock
- Solution: ???

tempuhs:

Arranging and presenting time



IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.

tempuhs:

Arranging and presenting time

- Using a universal clock we could index all events in relation to each other trivially
- However, this is slightly impossible because of non-linearity, relativity, & so forth

tempuhs:

Arranging and presenting time

→ Alternatives:

→ Different query for each clock

+Sort of solves the problem

+Slightly possible

-Scales poorly

→ Do nothing

+Quite easy

+Very possible

+Scales well

-Sort of doesn't solve the problem

tempuhs:

Arranging and presenting time

- Convert everything to TT for indexing
 - +Quite easy
 - +Sort of solves the problem
 - +An adequate amount of possible
 - +Scales well for terrestrial observers
 - Terrestrial-centric
- Different index for each observer clock
 - +Same as the TT-route & not terrestrial-centric
 - Scales poorly
 - Kind of a PITA

tempuhs:

Arranging and presenting time

Where we'd like to be tomorrow

→ Modifications to clocks and time

tempuhs:

Arranging and presenting time

- Presently we need to specify a minimum & maximum for each end point of the observation
- This is naïve and borked

tempuhs:

Arranging and presenting time

- Here's why it's borked
- We want to say “it happened in YYYY sometime”, and not have to deal with an artificial way of arriving at this
- Uncertainty cannot be expressed in a different time unit than the time unit used for the observation itself

tempuhs:

Arranging and presenting time

- Here's why it's borked
- Not very intuitive or flexible for a frontend engineer to implement
- Half the time it's not even useful/necessary
- Unmaintainable — what if a timespan's calendar changes

tempuhs:

Arranging and presenting time

- Solution: Model uncertainty and observation end points separately
- Problem: We need the unicorn format
- But let's pretend the problem doesn't exist for now

tempuhs:

Arranging and presenting time

- Observation end points are input as usual
- Uncertainty may be specified in various ways (including... not)
- “It happened sometime last year” becomes semantically distinguishable from “it lasted from 15 to 16 o'clock”... hopefully

tempuhs:

Arranging and presenting time

Where we'd like to be tomorrow

→ **Weight & context: do u even liftM?**

tempuhs:

Arranging and presenting time

- Two million things happen at X, what is displayed and what isn't?
- We need a way of expressing how valuable it is to display a certain timespan

tempuhs:

Arranging and presenting time

- Presently we express it through a manually input weight
- This is broken in ways we don't even have to explain

tempuhs:

Arranging and presenting time

- We need to be able to compute a valuableness
- The valuableness factor is different for every user for every context
- Something something research team 5 years something something

tempuhs:

Arranging and presenting time

- We can approximate a solution via user-provided tags & space information that we warp into our time dimension
- ... We just don't know, OK?

tempuhs:

Arranging and presenting time

Where we'd like to be tomorrow

→ Relationships are hard /wrist

tempuhs:

Arranging and presenting time

- Per now we only have basic inheritance to aid us in expressing timespan relations
- This is cute, but not expressive enough

tempuhs:

Arranging and presenting time

- Problems include:
- Difficult to express precisely how/why the timespans are related
- Difficult (computationally as well as conceptually) to arrive at obscure relations

tempuhs:

Arranging and presenting time

- Problems include:
- A timespan cannot be inside of multiple timespans
- Timespans cannot be partially inside other timespans

tempuhs:

Arranging and presenting time

- Solution: BRB gonna read some papers on data modelling
- P.S. Our present model is Good Enough™ for now, don't worry about it, we'll get to it eventually

tempuhs:

Arranging and presenting time

Where we'd like to be tomorrow

→ **Recurring events**

tempuhs:

Arranging and presenting time

- Presently, events begin and end
- However, some events don't end; they recur forever
- & some events recur temporarily & sporadically
- & some events recur at different times

tempuhs:

Arranging and presenting time

- We can sort of express this now by saying that events end at Infinity
- But that's artificial & convoluted
- So what should we do instead?

tempuhs:

Arranging and presenting time

- The big challenge here is addressing calendar-specific problems without calendars
- Optimally one should be able to implement a calendar using timespans – the reverse dependency should not exist!

tempuhs:

Arranging and presenting time

- The solution likely pertains a new concept entirely – a cycle, which is a cyclical timeline – we think, maybe
- ... We just don't know, OK?

tempuhs:

Arranging and presenting time

Where we'd like to be tomorrow

→ **Dealing with calendars and “real-world”-problems**

tempuhs:

Arranging and presenting time

→ Fuck calendars

→ Seriously

→ Fuck'em

tempuhs:

Arranging and presenting time

- Problem: calendars are just random ad-hoc crazy things made by crazy people
- Our ways of expressing time as measured by a clock (i.e. timezones and the like) one-ups calendars by being too irrational to be dealt with sanely

tempuhs:

Arranging and presenting time

- Solution: There is none
- Pain o'hoy: each system needs to be dealt with separately
- Let's start with the Gregorian calendar and take it from there

tempuhs:

Arranging and presenting time

Where we'd like to be tomorrow

→ **The role of roles & mission of permissions**

tempuhs:

Arranging and presenting time

- This is how we think roles could work
- Every user gets a role with the same name as their username
- They are also added to a User role
- An administrator or moderator is added to a Moderator role

tempuhs:

Arranging and presenting time

- Timespans have per-role permissions
- Read – May view a timespan
- Write – May change a timespan
- Share – May copy a timespan
- Own – May change permissions

tempuhs:

Arranging and presenting time

- Friends systems and similar may be implemented using roles
- If user Foo befriends user Bar, Bar simply gets the friend-of-Foo role

tempuhs:

Arranging and presenting time

- Default permissions need to be in place for a new timespan
- If user Foo authors a private timeline, it is sane to say that the Foo role gets the True product
- Everybody else gets the False sum

tempuhs:

Arranging and presenting time

- But if the user says “this may be viewed by my friends, the friend-of-Foo role takes on { own = False, read = True, write = False, share = False }
- And if it's a public timespan, the User role may take on { own = False, read = True, write = False, share = True }

tempuhs:

Arranging and presenting time

- A user likely has several roles
- A boolean sum dictates their actual permissions

tempuhs:

Arranging and presenting time

- This probably makes roles & permissions highly configurable
- Even to the point of being completely optional
- The semantics of the roles are completely left to the library users

tempuhs:

Arranging and presenting time

- N.B. We just made this up
- Hopefully it works
- It seemed like a good idea at the time

tempuhs:

Arranging and presenting time

- Srsly though
- The code is good
- We can fix this
- Trust us mk

tempuhs:

Arranging and presenting time

- <https://secure.plaimi.net/works/tempuhs.html>
- <https://github.com/plaimi/tempuhs>
- <https://github.com/plaimi/tempuhs-server>