Formal VINDICATIONS
Verificación Formal de Especificaciones Técnicas y Software

GURETRUCK

Guillermo Errezil Alberdi

Technical director, Formal Vindications S.L. & CEO, Guretruck S.L.

21 OCTOBER 2020

UNIVERSITAT DE BARCELONA

GURETRUCK

FVTM FV Time Manager powered by Formally Verified Time Library

EUROPEAN UNION
EUROPEAN REGIONAL DEVELOPMENT FUND

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

Bosch i Gimpera
UNIVERSITAT DE BARCELONA

DOCTORATS INDUSTRIALS
talent alliance!

Generalitat de Catalunya

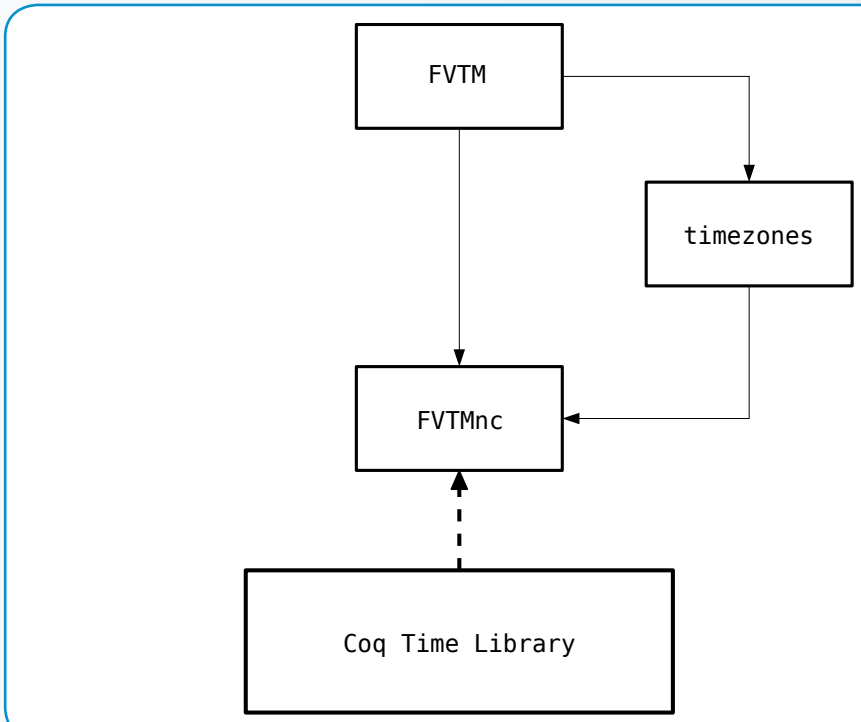
What is FVTM?

FVTM

- FVTMnc.ml;
- FVTMnc.mli;
- FVTM.ml;
- timezones.ml;
- Coq Time Library.

Pure version files	*Impure version files*
FVTMnc.ml FVTMnc.mli	FVTM.ml timezones.ml

Flowchart of FVTM file dependencies.



The technical specifications PDF can be used by:

1. The general public to understand the evolution of calendars throughout history to nowadays, with the UTC calendar with leaps seconds when it was discovered that the rotation of the earth is not constant.

2. Engineers to understand how to use the functions and their functionalities, and why it was built this way.

3. Expert mathematicians in this field to use as a guideline to understand the Time Library within the scientific community context.

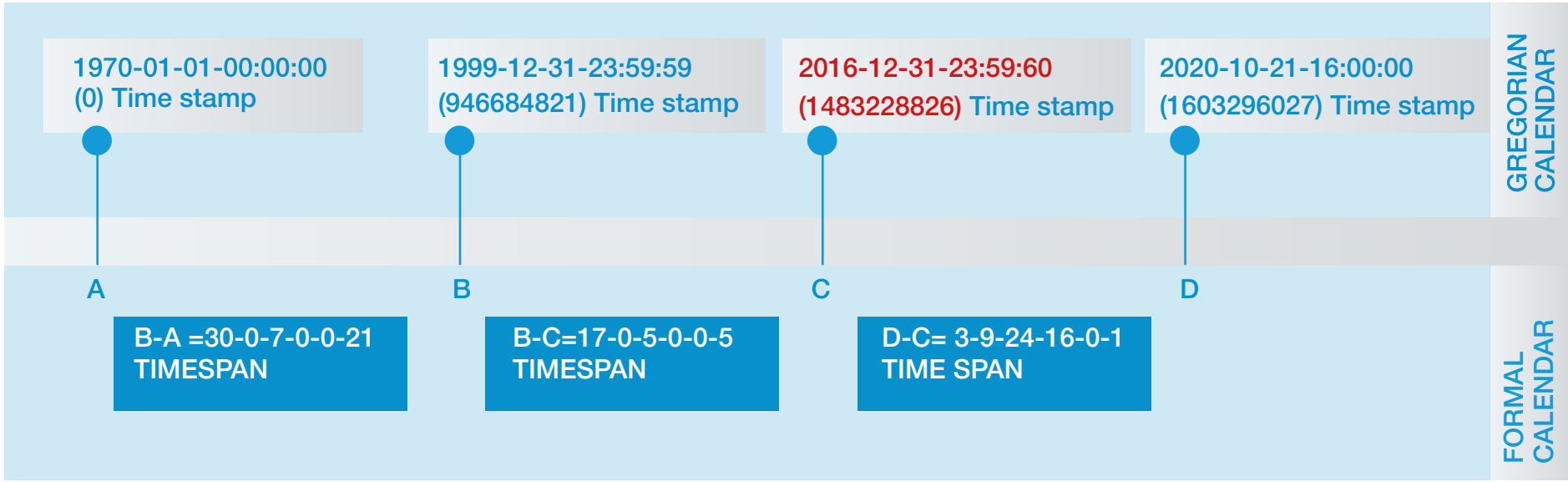
Main features FVTM

1 - Fully UTC, Date and Timestamp.

- **Timestamp:** Atomic seconds since UTC start point 1970-01-01 00:00:00, including leap seconds.

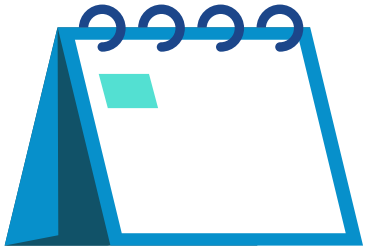
2 - Formal time and formal calendar. Constant ways of measuring time duration.

- Since minutes are not constant in UTC atomic clocks, we offer the solution by grouping seconds of any timespan (duration of an interval), in common unix based timespan the day the maximum group the seconds can be grouped.



THE FORMAL MINUTE DURATION WILL BE 60 ATOMIC SECONDS

THE FORMAL HOUR DURATION WILL BE 60 FORMAL MINUTES



THE FORMAL DAY DURATION WILL BE 24 FORMAL HOURS

THE FORMAL MONTH DURATION WILL BE 30 FORMAL DAYS

THE FORMAL YEAR DURATION WILL BE 365 FORMAL DAYS,

That is 12 formal months duration plus 5 formal days duration, which is the same as 31.536.000 seconds = $3{,}1536 \quad 10^7$

3 - Shift function versus addFormal function, critical for the legal world.

- Shift functions (shift seconds, shift minutes, shift days…) are widely used in most commercial Time Managers. Due to their common use, we keep them in UTC. However, they are very risky, terrifying in the legal world.

Our proposal is to substitute them by the addFormal functions.

Let's check an example:

ShiftUTCMonths	AddFormalMonths
2020/03/31/00:00:00	2020/03/31/00:00:00
+ 1 month	+ 1 formal month
2020/04/30/00:00:00	2020/04/30/00:00:00
-1 month	-1 formal month
2020/03/30/00:00:00	2020/03/31/00:00:00

4 - The addFormal functions have good arithmetical properties.

Given A1, A2 two time objects **(A2 > A1)**, if **D** = **timeDifference A2 A1**,
then our functions addFormal and substractFormal are consistent:
A2 = addFormal A1 D and **A1 = subtractFormal A2 D**.

5 - Calculations can only be made in UTC, we took the decision not to allow this in local time.

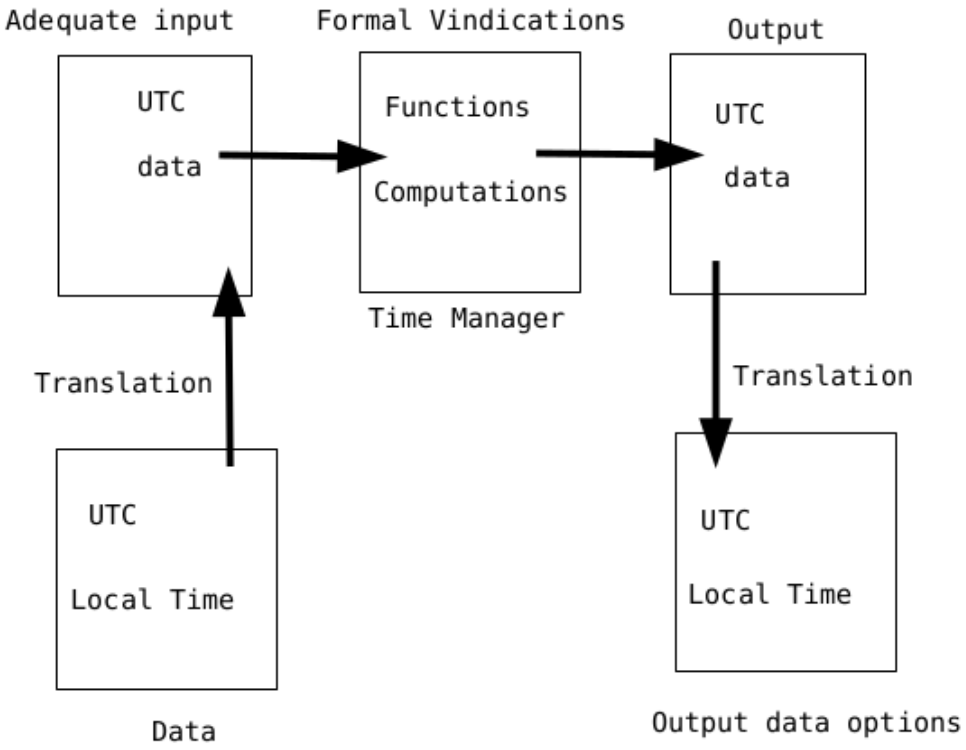


Figure 1: Behavior of the **FVTM**

- Are they the same? No, not at all and can be very dangerous

  Let´s check this example:

activity	time	kind
Break/Rest	2018/10/27/23:30:00	UTC
Driving	2018/10/28/00:55:00	UTC
Break/Rest	2018/10/28/01:05:00	UTC
Work	2018/10/28/22:00:00	UTC

- Driving time is 10 minutes in UTC

  If we do the calculation in Europe/London local Time (knowing that was hour change that day)

activity	time	kind
Break/Rest	28/10/2018/00:30:00	London local time (UTC+1)
Break/Rest	28/10/2018/01:05:00	London local time (UTC+0)
Driving	28/10/2018/01:55:00	London local time (UTC+1)
Work	28/10/2018/22:00:00	London local time (UTC+0)

  Driving time is 20 h 5 minutes making calculation in London/ Europe timezone

6 - The distance that light travels in a formal year could be a consistent unit of length.

- If we approximate the light speed by 3.10^8 m/s.
- Light-formal year = $9.1536 \bullet 10^{15}$ m, the distance that light travels in a formal year.

7 - Formally verified sorting of numbers!

Does anyone know a software where the numbers are sorted and there is a mathematical proof that it is correct?

Probably for commercial use it is not necessary, however if it is used by ESA or NASA in a rocket I would do it.

This is only to measure the level of our technology.

Issues to reach EAL8

- We could define EAL8 like totally formal verified proccess, no single step not formally verified.

1- Interpretation concept

A Only a specification made in mathematical language can be formally verified; however, it is very complex to understand with “words”. ***Then what are we formally verifying?***

B A specification made in words and understandable cannot be formally verified, because in general we cannot proof the correspondence of words and consistent mathematics.

Issues to reach EAL8

2 - The extraction from the proof assistant (COQ) to the software (OCAML) should be formally verified

One of the most problematic parts is that the proof assistants accept the natural numbers, (1,2,3..), but the software doesn't accept natural numbers, they only accept integers.

There is no computer powerful enough in the world to do this: 1 million +1 based in natural numbers .

Issues to reach EAL8

3 - The input control

Directly extracted code does not perform proper testing of any input.

The input expected to be perfect to have guaranteed results.

This 3 is related to point 2.

Our future work regards the possibility of providing mathematical proof of the equivalence of a bounded fragment of nat (1,2,3,,) and the non-negative fragment of the integers (1,2,3..) , in such a way that we would solve two problems at once: first, we would avoid extracting from the unbounded type nat to the bounded type int[4]; and second, we would be able to control inside of Coq the behavior of the functions when negative inputs are given.