



G-Nut library developments *relevance to EUREF Working Groups*

J. Douša, P. Václavovic, G. Györi

[\(jan.dousa@pecny.cz\)](mailto:jan.dousa@pecny.cz)

EUREF TWG – spring meeting (March 21-22, 2013, Brussels)

G-Nut core library

- **G-Nut project – development of generic library and end-user applications**
- **C++ and OO design and features**
- **Multi-threading and multi-platform applications**
 - currently Linux supported, but Windows and others using Boost's threads and asio libraries
- **Aimed for multi-GNSS: GPS, GLONASS, Galileo, BeiDou,**
- **Adjustment methods: LSQ, filtering (Kalman, SRCF), back-smoothing**
- **Processing techniques: real-time, near real-time, offline processing**
- **Initial target end-user applications:**
 - Troposphere monitoring
 - Positioning
 - Data monitoring, editing, splicing
- **Supposed to contribute to EUREF's WG and other projects:**
 - EUREF: Multi-GNSS, Real-time, Re-processing, Troposphere
 - E-GVAP, GNSS4SWEC, IGS, CzechGeo/EPOS, ...
 - ... but also to support other users to contribute ...

Specific applications & availability

Applications developed (and to be finalized for a release)

- **Geb** – PPP-client for RT/PP positioning (filtering/batch processing)
- **Tefnut** – PPP-based client for RT/NRT/offline troposphere estimation
- **Anubis** – QC monitoring, observation and nav message editing/splicing
- **NtripTools** – NTRIP client, NTRIP server, NTRIP pipe

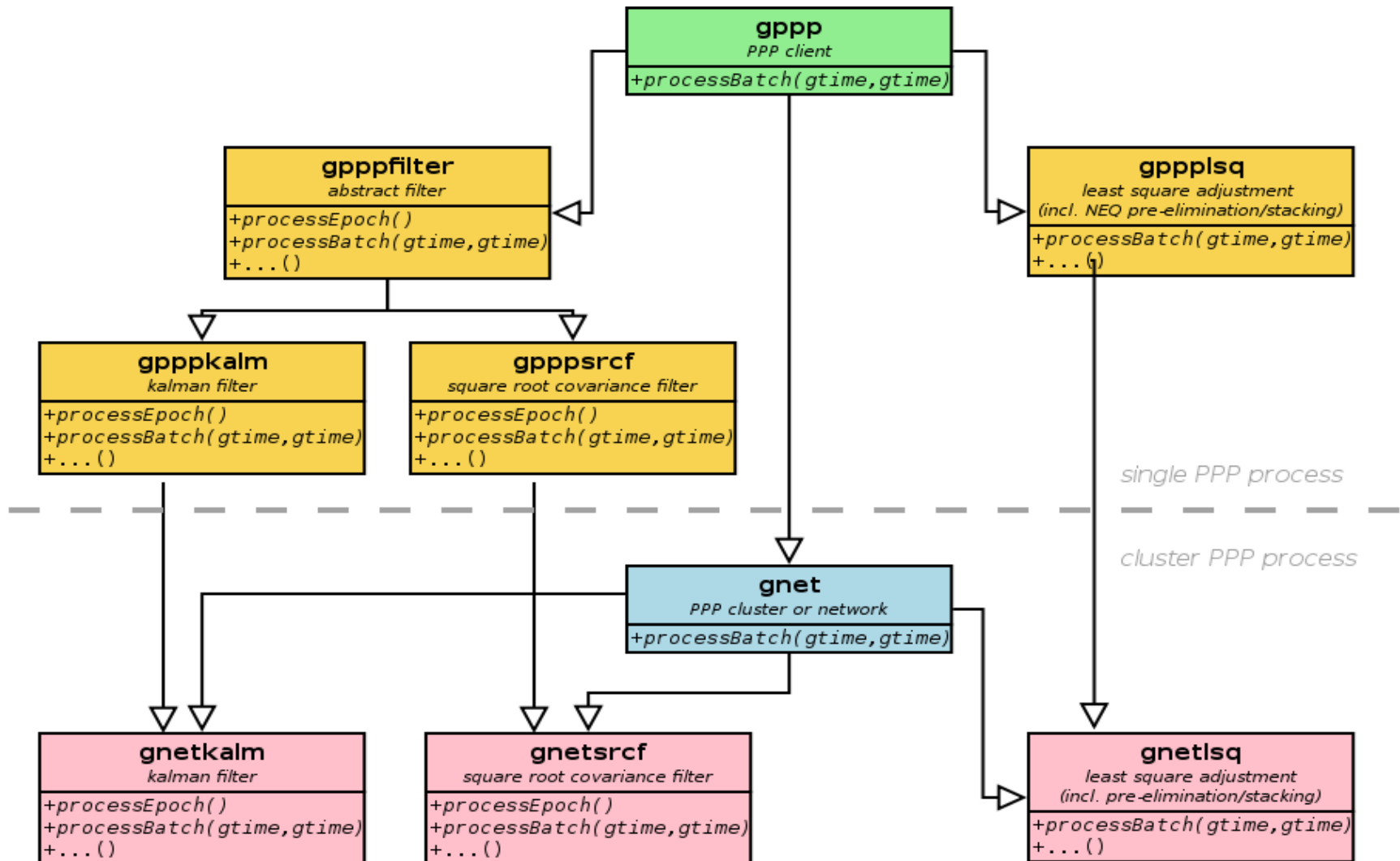
.... and others ...

Expected availability:

- priority to provide a full open-source and to support mainly scientific use
- G-Nut core library distributed only along with the specific applications
 - Provided as an application-specific package
- expected light/enhanced versions of application functionalities
- currently beta version delayed (lack of manpower for preparing release ...)

Processing / adjustment techniques

- Implementations of a single PPP process (gppp) and PPP cluster (gnet)
- Inheritance of adjustment methods for both processing models



Multi-GNSS observations & navigation

- GNSS navigation messages using OO inheritance and polymorphism
gnav - gnavgps, gnavglo, gnavgal, ... (available in a common container)
- RINEX3-like observations type internal definitions, implemented via enumerate type
-> flexible, expandable, not limited to char[3]
- Pre-defined priority table for various linear combinations (or user-defined) exploiting different bands and code/phase observations

PhaseLC(c1,c2,c3) with $LC = c_1.b_1 + c_1.b_2 + c_3.b_3$ ← PhaseObs(band,obs-type)

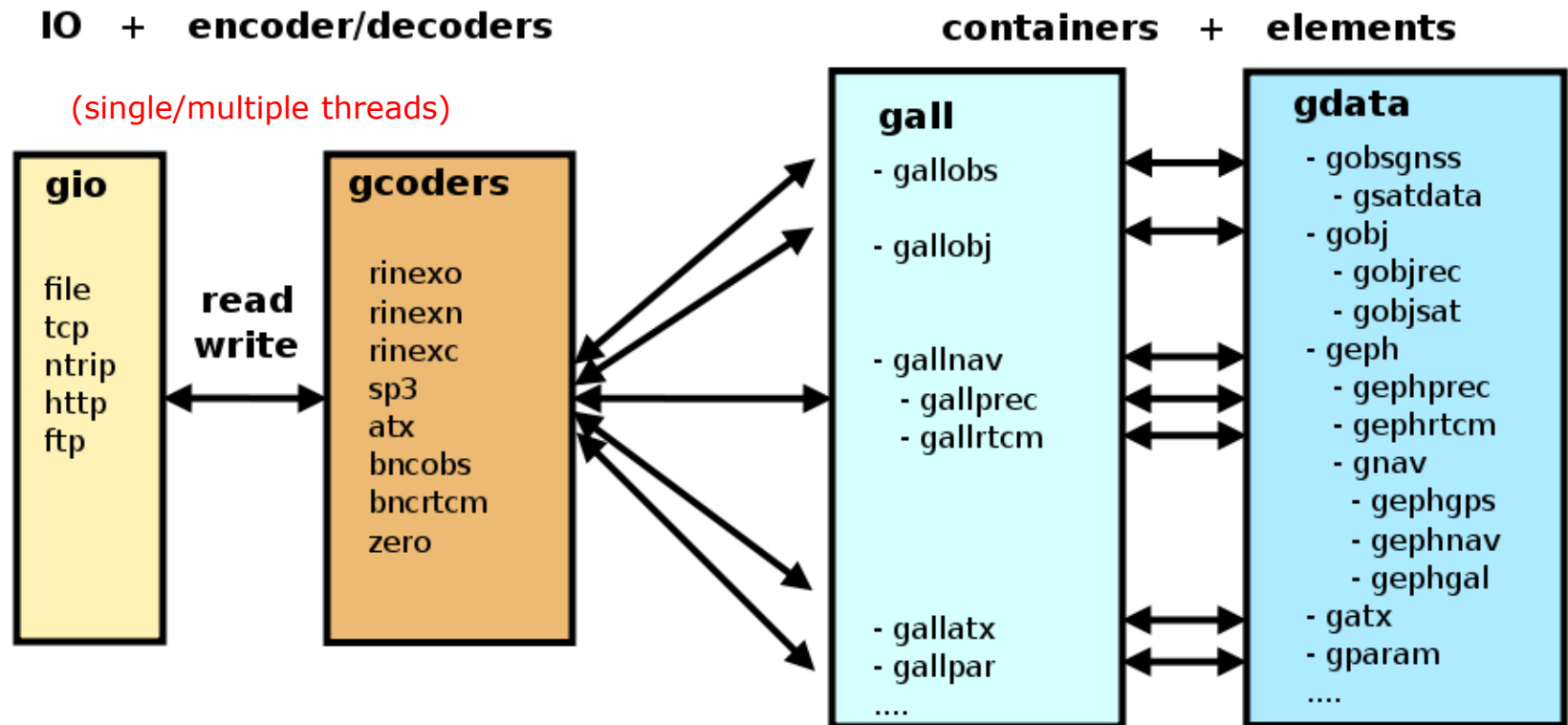
IonoFree(b1, b2, b3), GeomFree(b1, b2, b3), NarrLane(b1, b2, b3), WideLane(b1,b2,b3), MW...

		CHANNEL / CODE																	
FREQUENCY -- BAND	GPS NAVSTAR		GLONASS		Galileo		GLONASS		Galileo		GLONASS		Galileo		GLONASS		Galileo		
	C1A	C1B	C1C	x	x	C1L	C1M	x	C1P	x	C1S	C1W	C1X	C1Y	C1Z	P1	C1	CA	CB
	x	x	C2C	C2D	C2I	C2L	C2M	x	C2P	C2Q	C2S	C2W	C2X	C2Y	x	P2	C2	CC	CD
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	x	x	x	x	C5I	x	x	x	x	C5Q	x	x	C5X	x	x	P5	C5	x	x
	C6A	C6B	C6C	x	C6I	x	x	x	x	C6Q	x	x	C6X	x	C6Z	x	C6	x	x
	x	x	x	x	C7I	x	x	x	x	C7Q	x	x	C7X	x	x	x	C7	x	x
	x	x	x	x	C8I	x	x	x	x	C8Q	x	x	C8X	x	x	x	C8	x	x
	L1A	L1B	C1C	x	x	L1L	L1M	L1N	L1P	x	L1S	L1W	L1X	L1Y	L1Z	x	L1	LA	LB
	x	x	C2C	L2D	C2I	L2L	L2M	L2N	L2P	L2Q	L2S	L2W	L2X	L2Y	x	x	L2	LC	LD
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	x	x	x	x	L5I	x	x	x	x	L5Q	x	x	L5X	x	x	x	L5	x	x
	L6A	L6B	C6C	x	L6I	x	x	x	x	L6Q	x	x	L6X	x	L6Z	x	L6	x	x
	x	x	x	x	L7I	x	x	x	x	L7Q	x	x	L7X	x	x	x	L7	x	x
	x	x	x	x	L8I	x	x	x	x	L8Q	x	x	L8X	x	x	x	L8	x	x
GPS NAVSTAR				GLONASS				Galileo				COMPASS							

Real-time / offline support

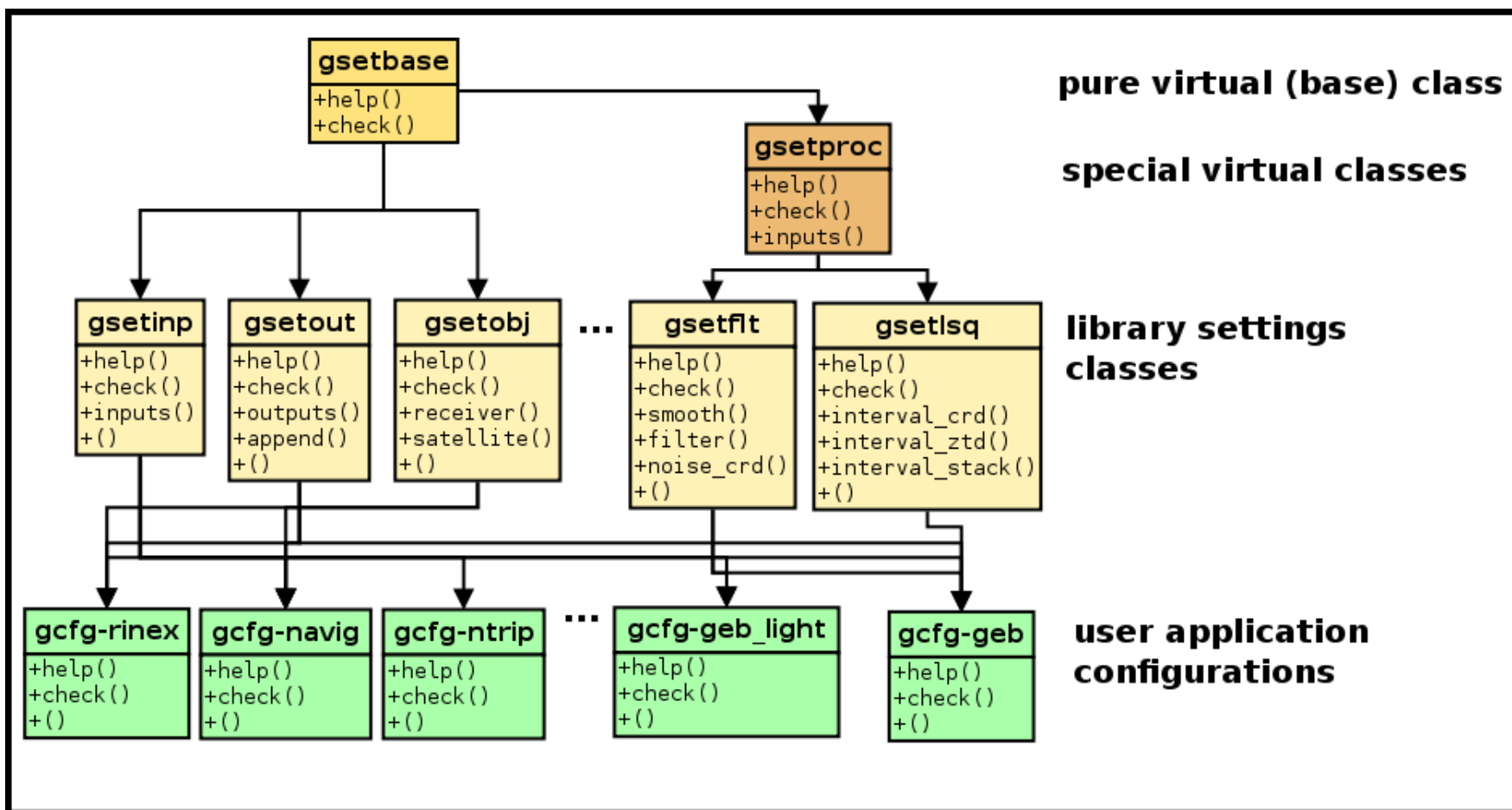
- Flexible OO input/output uniquely supports FILE, TCP, NTRIP (i.e. streams & files)
- Flexible implementation of encoders/decoders using internal buffer (ASCII/BINARY)
Generic base classes: `read_init()`, `write_init()`, `read_head()`, `read_data()`, `write_head()`, `write_data()`...
- Data and products storage based on flexible & expandable containers collecting self-consistent data elements (usually based on standard library map containers)

Content can be handled for: `1-epoch/N-epochs/fromTime - toTime/...`



Library & applications configuration

- XML configuration based on PUGIXML to support command-line & GUI settings
- Flexible implementation in two levels: 1) library classes, 2) end-user applications
- Regular/on-the-fly reconfigurations, embedded help, pre-defined black-box setting



Some more detail information

www.pecny.cz/GNSS/Software, submitted manuscript, presentations, ...



MAIN MENU

- Home
- GNSS
 - software
 - observations
 - data center (EUREF)
 - troposphere (E-GVAPII)
 - precise orbits (IGS)
 - analysis centre (EUREF)
 - reprocessing (EUREF)
 - EUREF-Czech-2009
 - projects
- DORIS
- Gravimetry
- Database
- Links
- Search

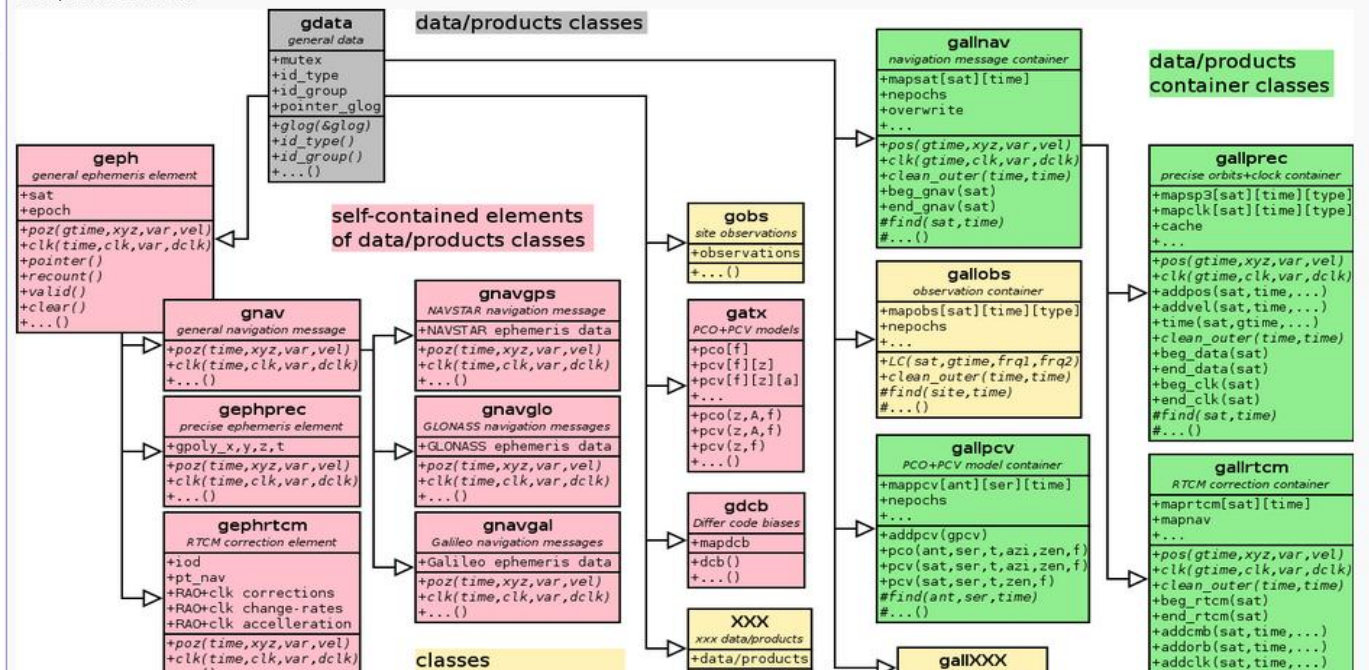
Page 3 of 4

Data, models, products self-contained classes and their containers

The main virtual base **gdata** class (dark grey background) represents any data, model or product classes either as self-contained data/product elements or their containers. This class provides a common mutex, **glog** pointer for common and multi-threaded logging and data type or group identification, which is later defined in each derived class.

Self-contained data/products elements (pink backgrounds) provides independent data such as e.g. all observation for a single station, satellite navigation message, RTCM position corrections, polynomials of precise ephemerides valid over a specific time, etc.

The containers (green backgrounds) are usually apply maps defined in a way to easily find the relevant self-contained element (pink backgrounds). In some classes (e.g. **gallnav**, **gallprec**) the cache is implemented to speed up the searching procedure, which is always done through an internal (**find**) function returning a pointer to specific data/product element.



Article Index

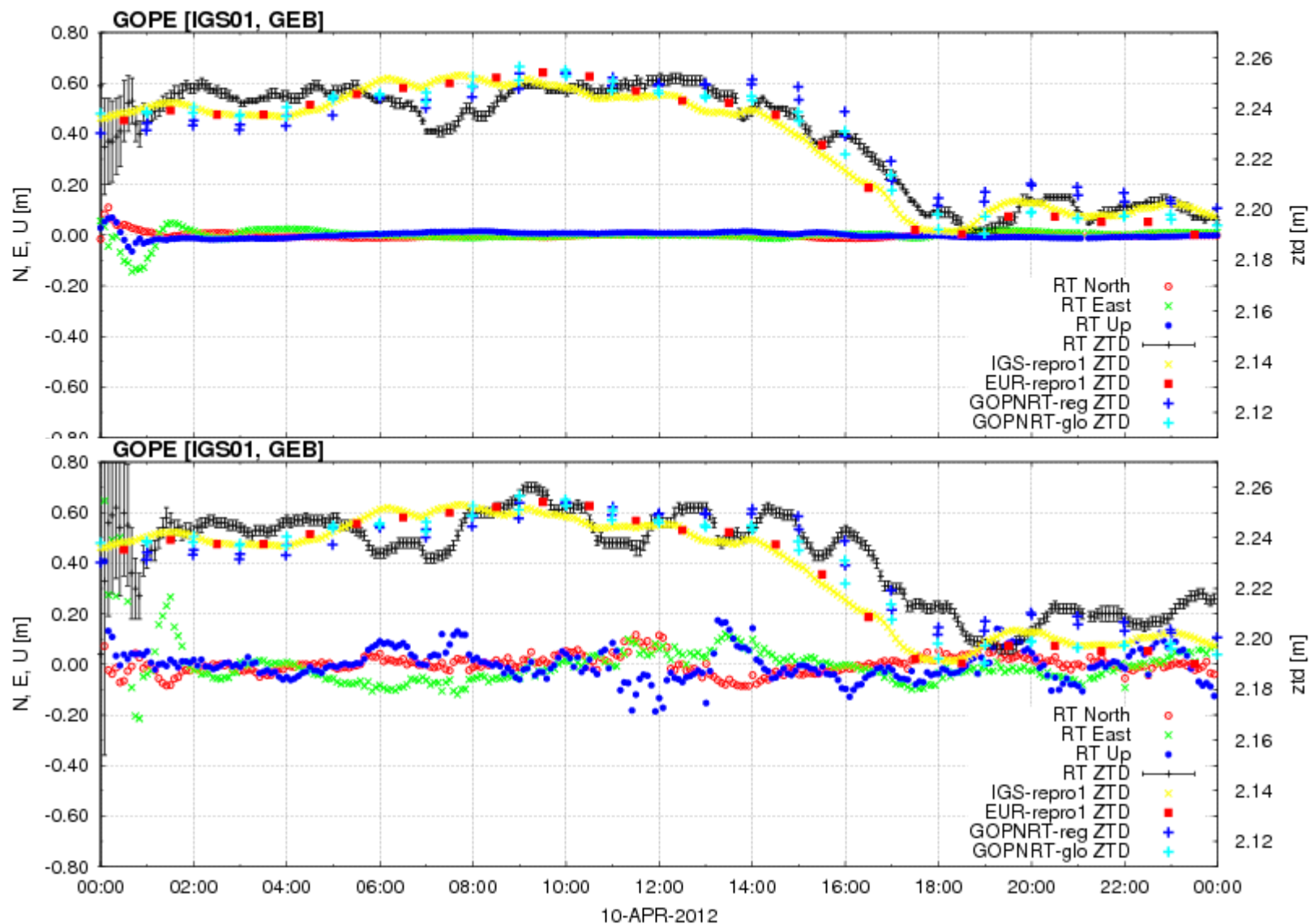
G-NUT - introduction
IO structure
Data structures
Applications

Anubis application

- Alternative to TEQC, BNC and other monitoring services
 - Editing: data, metadata, navigation messages editing/filtering
 - Splicing: data and navigation messages cutting/concatenating/filtering
 - Quantity: #obs/sats/bands/sys/gaps/ (using flexible internal containers)
 - Quality: ClockJumps, CycleSlips, MP, SNR, ... (using G-Nut preprocessing functions)
 - Outputs: using 1) data only or 2) data + navigation messages
 - ASCII logs only for easy grep and web-browsing
 - multi-verbosity in specific sections (metadata, quantity, quality, environment, ..)
 - + tools for visualizations (command-line or web-based or both)
- First focus on Quantity/Quality monitoring
 - Currently working on offline mode only, but easily transformable to (near) real-time mode
 - Monitoring GOP operational and data centre, maintaining historical data for Repro, ...
 - Supporting GOP analysis centre (Orbits, Repro, ...)
- Decoders/Encoders for observations & navigation messages
 - RINEX obs/nav decoders (v2,v3) – data tested on the MGEX campaign for all systems
 - RINEX obs/nav encoders to be developed later (to support editing, splicing)

Geb – coordinates & troposphere

Forward filter ZTD estimation using static/kinematic CRDs (April-May 2012)
using archived IGS01 RTCM corrections, example for site GOPE, day April-10



G-Nut on the web

<http://www.pecny.cz/> (GNSS, software) ... web-page with beta release soon



Main Menu

☐ Home

☐ GNSS

☐ software

☐ G-Nut library

☐ Geb (app)

☐ Tefnut (app)

☐ Anubis (app)

☐ observations

☐ data center

☐ troposphere

☐ precise orbits

☐ reference frame

☐ reprocessing

☐ EUREF-Czech-2009

☐ projects

☐ DORIS

☐ Gravimetry

☐ Observations

☐ Projects

☐ Database

☐ mail

☐ Links

☐ Theory

☐ mail-account

☐ formulars

☐ Search

☐ positions

GEB APPLICATIONS

Page 1 of 2

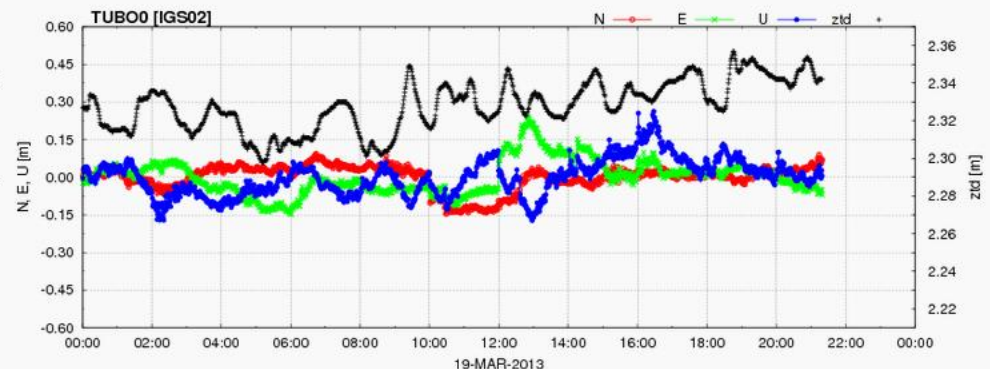
"Geb" are called end-user applications designed to estimate receiver position in a static, quasi-kinematic or kinematic mode. Below applications are implemented exploiting the **G-Nut core library** developed at the Geodetic Observatory Pecny. A comparison of supported capabilities is available in the **feature matrix**. All applications are released under the **GNU General Public Licence**. Figure shows an example of Geb's real-time coordinate monitoring in a quasi-kinematic mode using IGS RTCM clock and orbit corrections.

Geb-PP is an open source lite application aimed for the post-processing mode. A source code is available [here](#) and example configurations in the **support center**.

Geb-RT is an open source lite application aimed for the real-time processing mode. A source code is available [here](#) and example configurations in the **support center**.

Geb is an application providing enhanced capabilities while including all processing modes (RT and PP). If you find any of Geb lite applications useful in your work, consider its further development by ordering this advanced version (**under preparation**).

STATUS: the current lite versions are still provided as beta releases and still undergo final consolidations of the core library.



Geb applications
News
All Pages

Prev Next >>