International GNSS Service,
Analysis Center Workshop 2008
2-6 June 2008, Miami Beach, Florida, USA

Developing a Generic Multi-GNSS Software Package

Michael Meindl[a], Rolf Dach[a], Stefan Schaer[b],
Urs Hugentobler[c], Gerhard Beutler[a]

[a]Astronomical Institute, University of Bern, Switzerland
[b]Swiss Federal Office of Topography swisstopo, Switzerland
[c]Technische Universität München, Germany

Introduction

The Bernese GPS Software (BSW) was developed in the late 1980s. It was designed to process two frequency phase and code observation data from the Global Positioning System.

In 1999, GLONASS capability was added to the software by adapting and updating the already existing structures and algorithms.

With the upcoming new next-generation GNSS and the various new features and possibilities, such a straight-forward implementation in the software is not feasible, anymore.

A thorough software redesign is necessary.

Requirements for a multi-GNSS processing software

The various requirements for a GNSS software affect the principle software design. A multi-GNSS software should be, or should have:

- open for a wide variety of observation types (phase, code, SNR, doppler, inter-satellite, ranges, ...)
- process any desired number of frequencies
- easy and flexible handling of new linear-combinations of observations (with 2 or more components)
- be able to deal with different receiver tracking technologies (cf. observation codes defined in RINEX 3)
- correct handling of all relevant biases (DCBs, inter-system biases, ...)
- flexible and easy expandable file formats
- easy inclusion of future GNSS (e.g., Chinese COMPASS) and/or other space-geodetic techniques
- easy inclusion/adaption of new models and parameter types
- correct handling and combination of observations from single-GNSS and multi-GNSS receivers
- possibility of a fully consistent analysis of all involved GNSS on observation and NEQ level

Figure 1 gives an impression of the complexity of a multi-GNSS software showing the new observation selection panel of the BSW.

Basic design principles

Figure 2 shows the basic software architecture. The dashed box indicates the boundary between software internal and external world.

Modularization

All processing programs of the software depend on various core modules.

Basically, they contain all the GNSS definitions and information needed by the programs, such as carrier frequencies, observation types, linear combinations, etc (see Figure 3).

The module data/information can be accessed only via dedicated module functions. Carrier frequencies, linear combinations, etc. are not coded in the processing programs or sub-routines.

This modularization approach greatly facilitates new implementations (only core module must be updated) and helps keeping the main programs independent from external factors.

Independence from external formats

The processing programs are using only internal file formats and naming conventions.

All external files (RINEX, ...) are converted to internal formats by a collection of dedicated interface modules. The external nomenclature (e.g., RINEX 3 observation codes) is translated to a set of internal global identifiers defined in these interface modules.

Changes/updates of foreign file formats do not have any direct effect on the central processing programs. Therefore, only corresponding inter-face modules must be adapted.

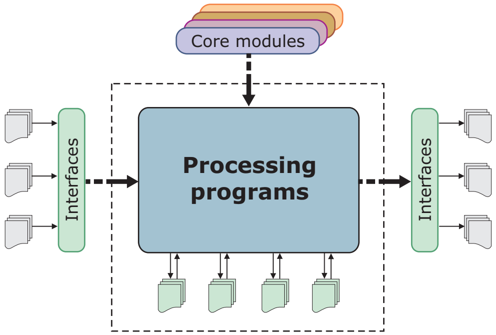

Figure 2: Software design diagram (parts inside the dashed box are shielded from the "outside" world).

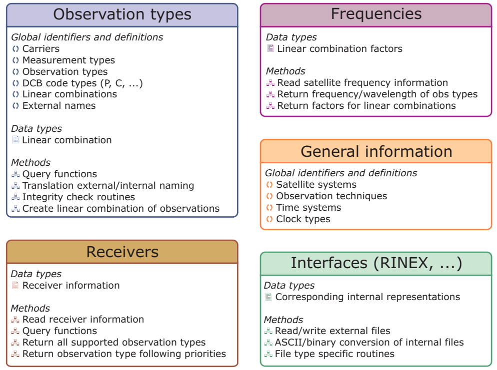

Figure 3: Configuration of the basic core modules and interfaces.

Flexibility and expandability

Special care is taken to ensure flexibility and easy expandability of the software package.

There are almost no hardwired limitations in the processing programs (e.g., maximum number of satellites or frequencies), anymore.

Internal file formats should be as flexible and generic as possible. ASCII type is preferred if file size and time critical access do not play an important role. Easy (human) readability should be maintained wherever possible.

Figure 4 shows the redesigned DCB file format of the BSW as one example of the new flexibility and openess. It offers a wide variety of bias types, freely selectable code types, optional time windows, and flags.

Summary

A multi-GNSS processing software should be flexible and easily adaptable to new observables, systems, and other space-geodetic techniques (such as SLR and VLBI).

The software should be highly modularized and independent from external formats and naming conventions. Main processing programs should communicate with the "outside" world using well defined interface routines, only.

For the redesign of the Bernese GPS Software about 40 from 100 programs, 600 from 1200 subroutines, 15 file formats, and most output files must be changed, updated, or rewritten.

The resulting time and effort for the necessary changes is about six man-years.

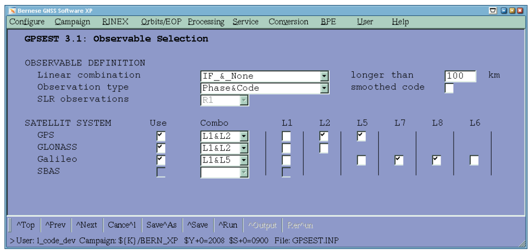

Figure 1: Observable selection for the new Bernese Software (draft).

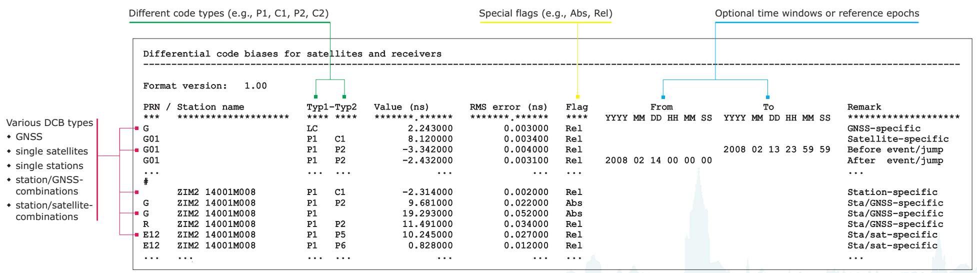

Figure 4: New DCB file format for the BSW.

AIUB
Poster compiled by M. Meindl, May 2008
Astronomical Institute of the University of Bern
michael.meindl@aiub.unibe.ch