Current Status of Development of New VLBI Data Analysis Software

Sergei Bolotin, John M. Gipson, David Gordon, Daniel S. MacMillan

NVI, Inc. 7257D Hanover Parkway Greenbelt, MD 20770 NASA Goddard Space Flight Center Greenbelt, Maryland 20771 USA

20th EVGA Meeting Bonn, Germany, March 29-31, 2011





VLBI data analysis software

New generation VLBI data analysis software

- In 2007 the IVS Working Group on VLBI data structures (IVS WG4) was established.
- In August of 2009 the VLBI group at the NASA GSFC started the development of new VLBI data analysis software.
- We presented the design and architecture overview of the new software has been presented a year ago at the IVS General Meeting at Hobart (Tasmania).
- In March. 2010 first lines of the source code were written.
- The first executable that will replace a part of current CALC/SOLVE system will be ν **Solve**, an interactive part of SOLVE System that is currently used in preliminary data analysis.
- In this presentation we will cover the current status of the software development process.





The software is written on C++ programing language.

It uses the $\ensuremath{\mathbf{Qt}}$ library for high level data abstraction and system libc, libm for low level system functions.

Currently, it consists of two parts:

- Space Geodesy Library, where all algorithms are implemented (90% of source code);
- an executable ν Solve a driver that calls the library and organizes work with an end-user (10% of source code).



B 🕨 🖌 B 🕨

The new VLBI data analysis software will be distributed in sources.

To make the software distribution portable we are using autoconf, automake and libtool packages. These packages are parts of GNU Build System. With this we can:

- create a highly portable software distribution;
- adjust the distribution to local needs;
- minimize end-user effort to compile the package, e.g. the simplest way to install the software:

user@host:~> ./configure && make && make install



글 🕨 🖌 글 🕨

Software development environment Modules Functionality

Integrated development environment

As a text editor we chose **Geany** software.

It has the following advantages:

- possess basic IDE features;
- depends on only few external packages;
- independent on distributives;
- small, lightweight and fast;
- compliant with make;

Geany is known to run under Linux, FreeBSD, NetBSD, OpenBSD, MacOS X, AIX v5.3, Solaris Express and Windows.

mbols Documents	SgVIbiAuxObservation.cpp 🕱 SgVIbiBaselineInfo.h 💢 SgVIbiBaselineInfo.cpp 💢 SgVIbiSessionUbilities.cpp	×
ministi Documents Spirlanamisti Spirlanamisti Disa Spirlanamisti Spirlanamisti Spirlanamisti Spirlanamisti Spirlanamisti Spirlana Spirlanaki Amaginamisti Spirlanaki Amagina Spirlanaki Amaginamisti Spirlanaki Amagina Spirlanaki Amaginamisti Spirlanaki Amagina Spirlanaki Amaginamisti Spirlanaki Amaginamisti Spirlanaki Amaginamisti Spirlanaki Amaginamisti Spirlanaki Amaginamisti Spirlanaki Amaginamisti Spirlanaki	<pre> Splitzbackbackbackbackbackbackbackbackbackback</pre>	×
SgVIbiSourceinfo.h SgVIbiStationinfo.h SgZenithDelay.cpp SgZenithDelay.h	Additional and a set of the additional and a set of the additional additadditional additional addite additional additional additional addi	lion •
Bibbook link: ar en Status Ibbook link: rani Ibbook link: (ad Compilation finis	, IbshibB a 6Pdealam3dretos o 0Pdealam3dretos o 6Pdealam3dreta 6 0Pdealam3dreta filos a 6Pdealam4dreta o 6Pdealam4 , IbshibB a IbshibB a 1996 ar 4£ In -a ", πIBB II" "IBB0 II") Ibsf Start 4E Int "IBB0 II" 4£ In -a ", πIBB II" "IBB0 II") Ibsf StartestMir,	nie -

Geany screenshot.



∃ >

Software development environment Modules Functionality

Documentation Generator

Doxygen is a documentation system for C, C++ and many other programming languages.

The software can automatically generate reference documentation in different formats:

- HTML;
- man pages;
- LaTex;
- RTF;
- PDF.

The documentation is generated directly from the sources, which makes it consistent with the current source tree.

			se Geodetic Library - Mozill			_ = × i		
Eile Edit View	Higtory Bookmarks Icols	Help						
🔹 💩 🔹 🚳 🔕 🊖 🐻 Sie //thema/sib/Projects/=NG=AbGPGeo/dec/Atmi/index.html 🔅 🔹 🚺 🖬 Google								
General Purpos	e Geodetic Library 🛛 🗇					-		
Ceneral Purpose Geodetic Library Class Me Class Me Class Me File Memb Prile Memb	Matrixen Terminal Second Secon	In Territory Ter	Allow ServiceRecord Cl ServiceRecord Cl Spantowski (Service Spantowski (Service Spantowski (Service Spantowski (Service ServiceRecord Service Micro ServiceRecord Service ServiceRecord ServiceRecord Service ServiceRecord ServiceRecord Service ServiceRecord ServiceRecord Service ServiceRecord ServiceRecord Service ServiceRecord ServiceRecord ServiceRecord ServiceRecord ServiceRecord ServiceRecord ServiceRecord S	ass Reference	SgtenServesHucentte	Bgöttsarveat		
Done						11		







∃ >

Software development environment Modules Functionality

Modular structure of the software



System Decomposition

To keep our system stable and flexible we designed it modular.

Module is a logical block of code that is loosely tied with other parts of the software.

Each arrow on the diagram represents a **dependency** or, in other words, provides information (types, function calls, constants).

Only main **dependencies** are shown on the diagram.

< 🗇 🕨

- E - E

< ∃ →

Software development environment Modules Functionality

Modular structure of the software



Logging Subsystem

A small module that implements logging of system functionality.

There are four log levels:

- ERROR
- WARNING
- INFO
- DEBUG

and a set of log facilities.

The Logging Subsystem can filter log messages by level and facility, display filtered messages and save them in a log file.

This module is used by all other modules.

< 注 > < 注 >

< 🗇 🕨



Software development environment Modules Functionality

Modular structure of the software



Mathematical Tools

A module that contains mathematical data structures and procedures.

The following classes are implemented:

- Vector and Matrix general classes of linear algebra;
- Specialized objects, like Symmetric Matrix or Upper Triangular Matrix, to optimize calculations;
- Geometrical classes, like Vector3D, Matrix3D, etc – to make transformations of coordinates in 3d space.

Also, optimized versions of matrix operations are realized in this module.

→ Ξ → < Ξ →</p>



Software development environment Modules Functionality

Modular structure of the software



Modeling Subsystem

The main purposes of this module are: manipulation of geodetic data, computation of theoretical values and partials.

The module describes the following data structures, phenomena and models:

- Epoch of observation and time interval;
- Identities and attributes of various objects: radio sources, stations and baselines;
- VLBI observation and auxiliary data;
- Ionospheric calibration;
- Refraction correction;
- Model of clock breaks;

< 🗇 🕨

Physical and geophysical constants.

→ Ξ → < Ξ →</p>



Software development environment Modules Functionality

Modular structure of the software



Estimator

The module is responsible for performing Least Square Estimation.

Classes that are implemented in this module:

- Partial derivative;
- Estimated parameter;
- Configurator of parameters that will be estimated;
- Estimator performs LSM estimation for lists of parameters.

Current realization implements only unbiased, session-wide parameters. Later we will add global and stochastic parameters.

→ Ξ → < Ξ →</p>



Software development environment Modules Functionality

Modular structure of the software



I/O Subsystem

The module represents operations of input/output and supports various data formats.

Currently, only Mk3 database handling is implemented. The subsystem consists from two parts:

- A decoder of Mk3 DBH format;
- A logical image data that are stored in a Mk3 DBH file.

The module allows to read and write Mk3 DBH files and modify their structure.

→ Ξ → < Ξ →</p>



Software development environment Modules Functionality

Modular structure of the software



Data Flow Management

The module that performs data analysis and describes models and system configuration which should be applied in the analysis.

Classes that are implemented:

- Configurator a class that describes models and algorithms;
- Task manager of a single session.

This module can be considered as a core of the software – it uses almost all other modules and realizes communication between different parts of the software.

< 注 > < 注 >



< 🗇 🕨

Software development environment Modules Functionality

Modular structure of the software



Graphical User Interface

The GUI module interacts with a user and visualize observations, solutions and auxiliary data.

Classes that are implemented in this module:

• A Plotting Subsystem;

< 🗇 🕨

- A tool for task configuration;
- Browsers of lists of stations, baselines and sources.
- etc.

This module uses widgets from Qt library.

< 注 > < 注 >



Software development environment Modules Functionality

Graphical user interface

Plotting Subsystem

Features of the plotting system:

- Plots data split by branches;
- Input data can be multidimension;
- Allows user to browse "everything vs everything";
- Interacts with user, allowing user to modify data.

The plotting system is universal and can be used in various applications.



A screenshot of the plotting system.





Current functionality

Currently, the software development process is in the intermediate stage and not all functions are realized.

The software can:

- Read VLBI observations in Mk3 DBH format;
- Display various information that were stored in the files;
- Evaluate ionospheric corrections;
- Estimate parameters of clock functions, zenith delays, stations positions and source coordinates;
- Allow a user to resolve manually ambiguities;
- Detect and take into account clock breaks;





Software development environment Modules Functionality

lonospheric correction

Ionospheric correction

Currently, ionospheric calibration is evaluated by CALC/SOLVE system on the stage of initial processing of VLBI observations. It is introduced at the Version 4 of Mk3 DBH files.

In new VLBI data analysis software we implemented the same algorithms for evaluating effective frequencies and ionospheric corrections for group delay, phase delay and phase delay rate.





10 / 12

Clock break correction

Clock break processing

To compensate a clock break, $\nu {\rm Solve}$ adds a step-wise linear function to the station clocks.

The following procedures are used to process a clock break:

- Detection of a clock break: determination of the event, station and epoch;
- Evaluation of clock break value;
- Compensate clock break in the analysis.

Also, software allows a user to add a clock break event manually.





Plans

A first public release will be made in the middle of this year.

Following functions need to be implemented before the public release:

- · Piecewise and stochastic estimation of clock parameters and zenith delays;
- Automatic ambiguity resolutions;
- Reweightening;
- Export observations into current CALC/SOLVE data structures.

Today during poster viewing we will have a demonstration of ν **Solve**.

Thank you for attention!



