Revitalizing a Golden Age Geodetic Software System

Thomas Knudsen, Kristian Evers (SDFE), *Jakob Jakobsen, Lars Stenseng* (DTU space)

The Geodetic Software System, originating at the Danish Geodetic Institute, was based on ideas from the 1950s, and found its first true incarnation in the Algol language and the custom built geodetic number cruncher GIER (figure left), from 1961.

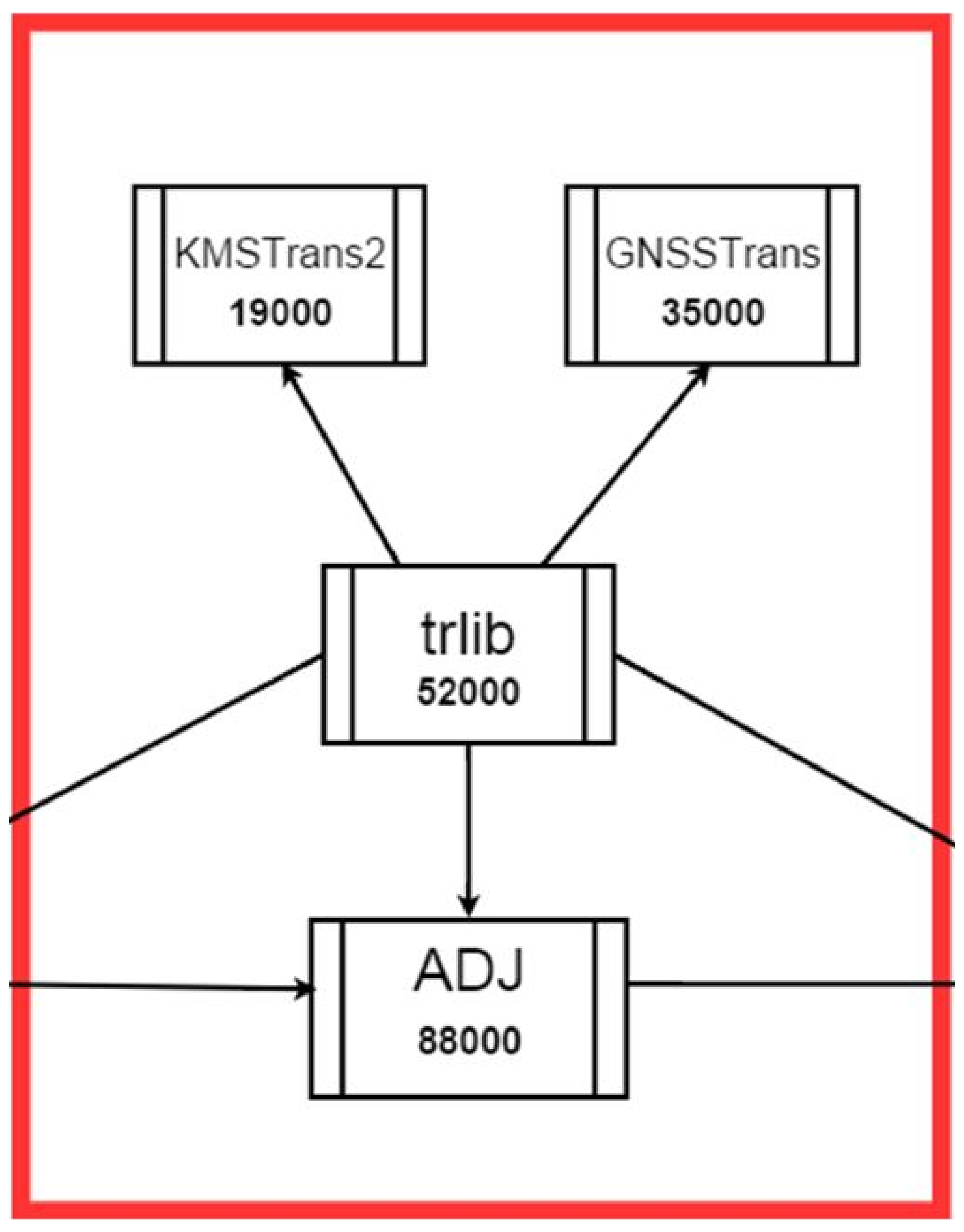
Through the years, the system evolved, and was ported to new programming languagues and computer architectures.

Since the 1990s, the system could broadly be perceived as consisting of 4 tightly coupled components

- A metadata management system
- A observation data base management system
- An adjustment system
- A transformation system

But the tight coupling, and the complex architecture resulting from 5 decades of development has made the system largely unmaintainable once the last of the original authors reached retirement.

Something drastic needs to be done...



System block architecture

The minimum viable part of the geodetic software system weighs around 200000 lines of code, distributed as shown left, between

- trlib, the transformation library
- ADJ, the 1D,2D,3D adjustment library
- KMStrans2, the GUI interface to trlib
- GNSStrans, the ETRS89 transformation package

But due to the tight coupling, the block delineations and the corresponding lines-of-code counts are somewhat arbitrary

Revitalization strategy

The primary strategic guideline for revitalization is

REDUCE
the amount of code to maintain locally by

DISCARDing
existing code

REIMPLEMENTing as needed
by contributing to exiting open source projects, hence

LEVERAGE
existing geomatics infrastructure environments (notably OSGeo)

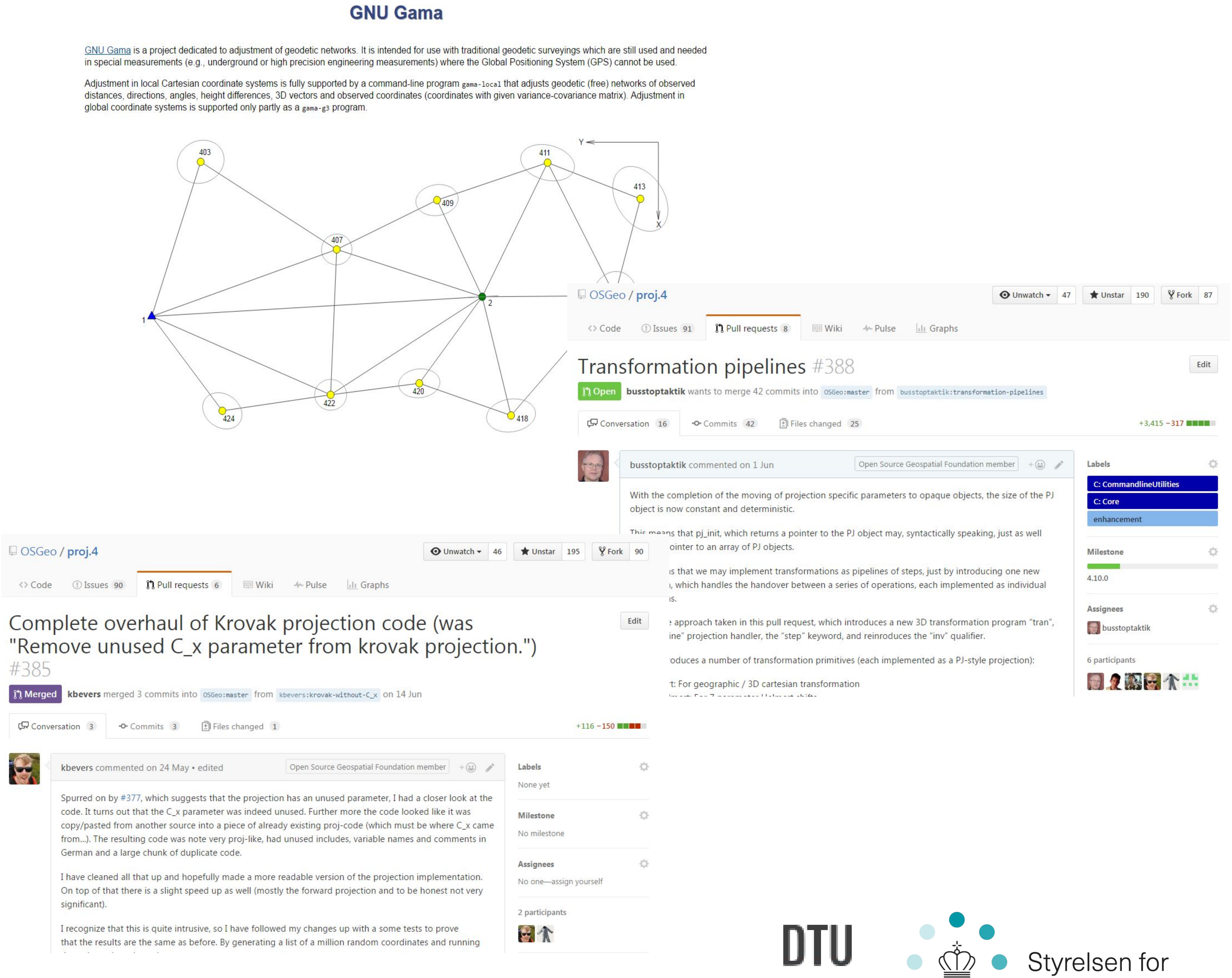
Implementation

At SDFE and DTU space, all GNSS adjustment has, for many years, been done using Bernese, by and large leaving only the (much simpler) 1D levelling adjustment unhandled. We intend to use the GNU Gama system for this.

Transformations and projections are reimplemented as part of the leading projection library proj.4 (originating at USGS, now part of the OSGeo METACRS project), which we co-maintain and contribute heavily to (cf. pull requests, right).

No final solution has been decided on for the **GNSS-specific transformation** production (i.e. conflating 14 parameter + deformation model to plain 7 parameter Helmert for a given location and epoch). **Any advice and suggestions would be most welcome**

The need for the **GUI interface to trlib** will be largely gone, once all national transformations are reimplemented in proj.4: At that time, they will auto***magic***ally be supported by the large majority of geospatial software, using proj.4 for its transformation functionality.



GNU Gama

GNU Gama is a project dedicated to adjustment of geodetic networks. It is intended for use with traditional geodetic surveyings which are still used and needed in special measurements (e.g., underground or high precision engineering measurements) where the Global Positioning System (GPS) cannot be used.

Adjustment in local Cartesian coordinate systems is fully supported by a command-line program `gama-local` that adjusts geodetic (free) networks of observed distances, directions, angles, height differences, 3D vectors and observed coordinates (coordinates with given variance-covariance matrix). Adjustment in global coordinate systems is supported only partly as a `gama-g3` program.

OSGeo / proj.4

Transformation pipelines #388

Open **busstoptaktik** wants to merge 42 commits into `OSGeo:master` from `busstoptaktik:transformation-pipelines`

Conversation 16 · Commits 42 · Files changed 25 · +3,415 −317

busstoptaktik commented on 1 Jun · Open Source Geospatial Foundation member

With the completion of the moving of projection specific parameters to opaque objects, the size of the PJ object is now constant and deterministic.

This means that pj_init, which returns a pointer to the PJ object may, syntactically speaking, just as well
ointer to an array of PJ objects.

is that we may implement transformations as pipelines of steps, just by introducing one new
, which handles the handover between a series of operations, each implemented as individual
is.

approach taken in this pull request, which introduces a new 3D transformation program "tran",
ine" projection handler, the "step" keyword, and reinroduces the "inv" qualifier.

oduces a number of transformation primitives (each implemented as a PJ-style projection):

t: For geographic / 3D cartesian transformation

Labels: C: CommandlineUtilities · C: Core · enhancement

Milestone: 4.10.0

Assignees: busstoptaktik

6 participants

OSGeo / proj.4

Complete overhaul of Krovak projection code (was "Remove unused C_x parameter from krovak projection.") #385

Merged **kbevers** merged 3 commits into `OSGeo:master` from `kbevers:krovak-without-C_x` on 14 Jun

Conversation 3 · Commits 3 · Files changed 1 · +116 −150

kbevers commented on 24 May • edited · Open Source Geospatial Foundation member

Spurred on by #377, which suggests that the projection has an unused parameter, I had a closer look at the code. It turns out that the C_x parameter was indeed unused. Further more the code looked like it was copy/pasted from another source into a piece of already existing proj-code (which must be where C_x came from...). The resulting code was note very proj-like, had unused includes, variable names and comments in German and a large chunk of duplicate code.

I have cleaned all that up and hopefully made a more readable version of the projection implementation. On top of that there is a slight speed up as well (mostly the forward projection and to be honest not very significant).

I recognize that this is quite intrusive, so I have followed my changes up with a some tests to prove that the results are the same as before. By generating a list of a million random coordinates and running

Labels: None yet

Milestone: No milestone

Assignees: No one—assign yourself

2 participants

DTU

Styrelsen for
Dataforsyning og
Effektivisering