Infrared Full Sky Scanning Mission as a Data Processing Challenge

Science and technology roadmap for μas studies of the Milky Way

Wolfgang Löffler

Astronomisches Rechen-Institut, Zentrum für Astronomie, Universität Heidelberg
loeffler@ari.uni-heidelberg.de

Lund: 18–20 July 2023

Wolfgang Löffler:

- Stellar Astrophysics
- Software Industry
- Astrometry Team at ARI/ZAH
- Software Architect for Gaia DPAC First Look
- Gaia DPAC One-Day Astrometric Solution (ODAS)
- Gaia DPAC First-Look Scientist
- ARI JASMINE Astrometric Solution (AJAS)

We often think that…

If we can write down the equations,
then we have understood the problem.

If we can write down the solution to the equations,
then we have solved the problem.

OK, let's see…

Astrometry: Observation Equation

Simple case: one model parameter, one observation:

$$o_\ell + \varepsilon_\ell = f_\ell(p) + r_\ell \qquad \text{with } p = \hat{p} + \Delta\hat{p}$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!} \frac{\mathrm{d}^n}{\mathrm{dp^n}} f_\ell(p)\Big|_{\hat{p}} (p - \hat{p})^n + r_\ell$$

$$\simeq f_\ell(\hat{p}) + \frac{\mathrm{d}}{\mathrm{dp}} f_\ell(p)\Big|_{\hat{p}} \Delta\hat{p} + r_\ell$$

o_ℓ observation, ε_ℓ observational error
f_ℓ model prediction, p model parameter, r_ℓ modelling error

Astrometry: Design Matrix

Realistic case: many model parameters, many observations:

$$\begin{pmatrix} o_1 \\ \vdots \\ o_\ell \\ \vdots \\ o_L \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_\ell \\ \vdots \\ \varepsilon_L \end{pmatrix} \simeq \begin{pmatrix} f_1(\boldsymbol{p}) \\ \vdots \\ f_\ell(\boldsymbol{p}) \\ \vdots \\ f_L(\boldsymbol{p}) \end{pmatrix}\Bigg|_{\hat{\boldsymbol{p}}} + \underbrace{\begin{pmatrix} \frac{\partial}{\partial p_1} f_1(\boldsymbol{p}) & \cdots & \frac{\partial}{\partial p_\lambda} f_1(\boldsymbol{p}) & \cdots & \frac{\partial}{\partial p_\Lambda} f_1(\boldsymbol{p}) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial p_1} f_\ell(\boldsymbol{p}) & \cdots & \frac{\partial}{\partial p_\lambda} f_\ell(\boldsymbol{p}) & \cdots & \frac{\partial}{\partial p_\Lambda} f_\ell(\boldsymbol{p}) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial p_1} f_L(\boldsymbol{p}) & \cdots & \frac{\partial}{\partial p_\lambda} f_L(\boldsymbol{p}) & \cdots & \frac{\partial}{\partial p_\Lambda} f_L(\boldsymbol{p}) \end{pmatrix}\Bigg|_{\hat{\boldsymbol{p}}}}_{D} \begin{pmatrix} p_1 - \hat{p}_1 \\ \vdots \\ p_\lambda - \hat{p}_\lambda \\ \vdots \\ p_\Lambda - \hat{p}_\Lambda \end{pmatrix} + \begin{pmatrix} r_1 \\ \vdots \\ r_\ell \\ \vdots \\ r_L \end{pmatrix}$$

$$\boldsymbol{o} + \boldsymbol{\varepsilon} \simeq \boldsymbol{c} + D\,\Delta\hat{\boldsymbol{p}} + \boldsymbol{r}$$

D design matrix

Astrometry: Normal Matrix and Solution

$$\boldsymbol{o} - \boldsymbol{c} = D\,\Delta\hat{\boldsymbol{p}} + \boldsymbol{r}'$$

Many more equations than model parameters
$\rightarrow$ Optimum least squares solution given by

$$\Delta\hat{\boldsymbol{p}} = (D^T D)^{-1} D^T (\boldsymbol{o} - \boldsymbol{c}) = N^{-1} D^T (\boldsymbol{o} - \boldsymbol{c})$$

$\Rightarrow$ "All we need to do" is to invert the normal matrix N.

$N = D^T D$ normal matrix, $\boldsymbol{r}' = \boldsymbol{r} - \boldsymbol{\varepsilon}$

Astrometry: Input Data (Example)

- Image Centroids:
  - fitting model PSF to light distribution
  - centroid location needed as input to select correct PSF

  $\longrightarrow$ Inner iteration

- Cross-Match:
  - star positions, attitude, calibration needed as input

  $\longrightarrow$ Outer iteration

Just preparing the astrometric input data
already involves complex data processing.

Astrometry: Data Processing

Solving the astrometric problem involves way more

Data Processing

than the simple equations want to make us believe.

Data Processing

Data: Past and Present

- Hipparcos
  - megabytes of data
  - core memory measured in kilobytes
  - external memory measured in megabytes – on magnetic tape
- Gaia
  - terabytes of data
  - core memory measured in gigabytes
  - external memory measured in terabytes – on magnetic hard disks

Hardware limits the amount of data that can be processed.
$\Rightarrow$ Clever data warehousing and pipelining necessary.

Data: Future

- JASMINE
  - terabytes of data
  - core memory measured in terabytes
  - external memory measured in terabytes – on solid state drives
- GaiaNIR
  - petabytes of data?
  - core memory measured in terabytes
  - external memory measured in petabytes – on solid state drives

Hardware limits no longer so restrictive.
$\Rightarrow$ New ways of data processing possible?

Data: Warehousing & Pipelining

Data Warehousing:

How to store data for quick and easy access.

Data Pipelining:

How to read/write data from/to external storage without slowing down the CPU/GPU.

Data Reduction:

Increasing the amount of data by orders of magnitude.

Data: Limiting Factor

Astrometry is not limited by CPU speeds.

Astrometry is limited by data storage size and data transfer speeds.

Data Processing: Process

Data Processing

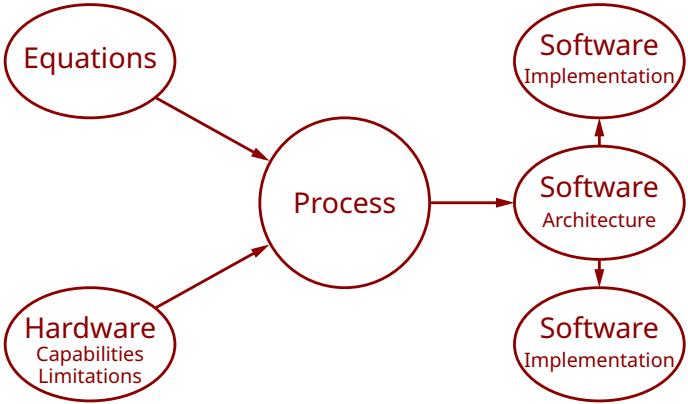
Process: Dataflow

- Ideally:
  Process = Dataflow through the equations.

- Realistically:
  Process = Dataflow along hardware capabilities
  and around hardware limitations.

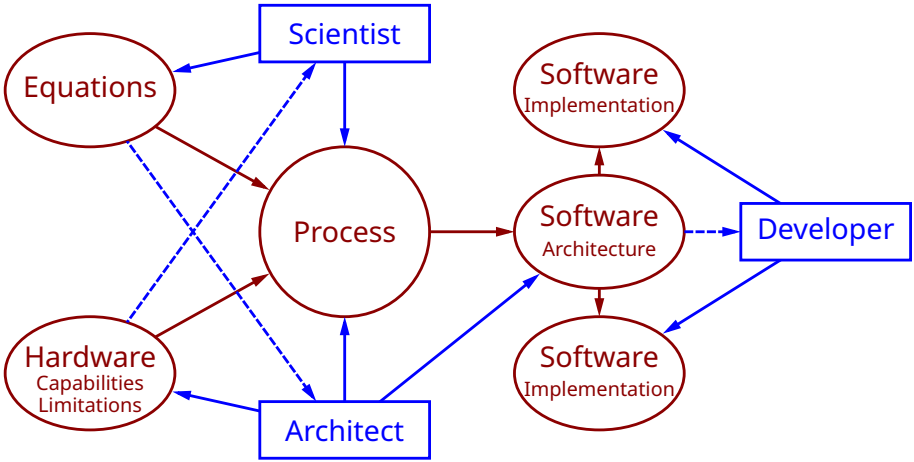
Pre-processing, side processing, post-processing, iterations.
→ The equations do not tell the full story.

Process: Software Implementation



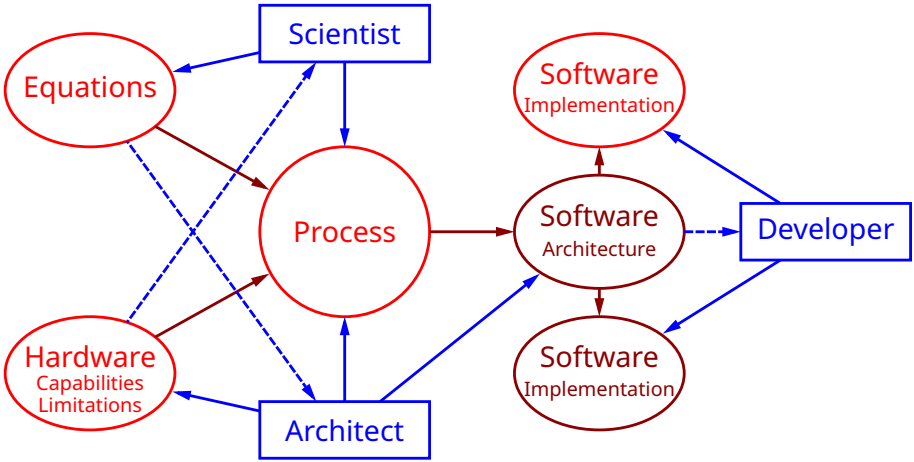
Nota Bene: Software is the implementation of a process.

Process: Roles



Nota Bene: One person may have more than one role.

Process: Evolution



Evolution of the process shall lead to minimal changes in software.

Process: Evolution, Minimal Changes

- People will gain knowledge and experiences.
- Equations will be added and changed.
- Hardware will become more powerful.

The process will be updated causing software changes.

$\Rightarrow$ The software architecture must be flexible enough to allow for updated processes.

$\Rightarrow$ The software implementation must be simple enough such as not to obfuscate the specific process.

Architect: Job Description

What we are looking for:

- Must understand the astrometric problem
- Must understand hardware capabilities and limitations
- Must understand scientists
- Must understand software developers
- Must have experience in process design
- Must have experience in software architecture

Architect: What we are looking for…

Architect: Job Description

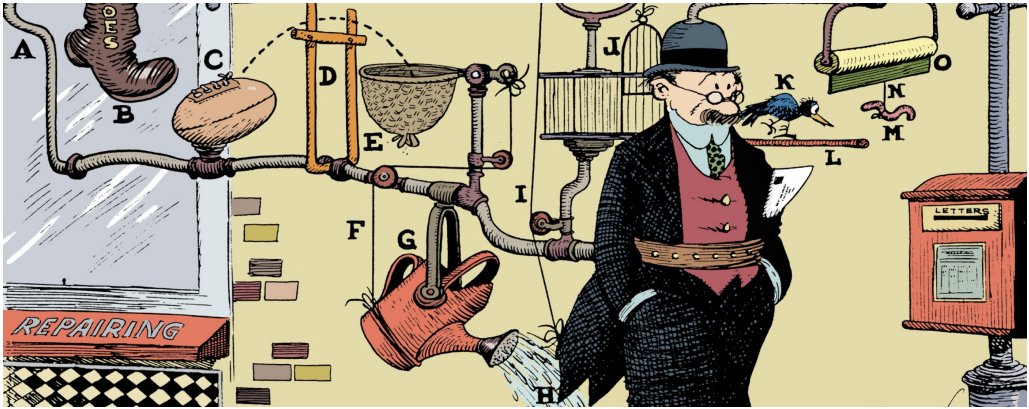
What we are offering:

- Interesting and challenging project
- Flexible working environment
- International collaboration
- Unattractive salary compared to industry
- Temporary position (even shorter than project duration)

Architect: What we might get. . .

Rube Goldberg Machine

What we want to avoid:



The Real Challenge

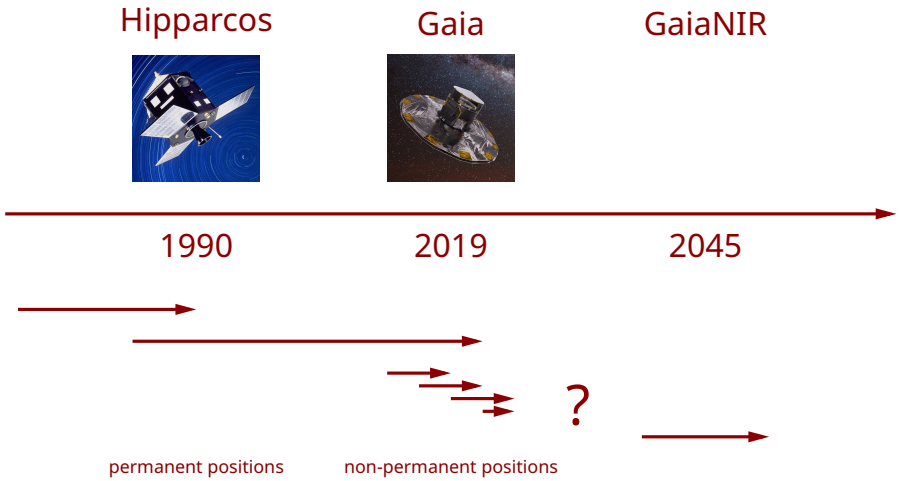
Astrometric Data Processing is foremost a

Human Resources Challenge.

We need not only excellent scientists
but also excellent process and software architects.

On permanent positions.

Carrying on the Flag and Bridging the Gap



Thanks for your Attention!

```
char*lie;
   double time, me= !0XFACE,
   not; int rested, get, out;
   main(ly, die) char ly, **die ;{
signed char lotte,

dear; (char)lotte--;
   for(get= !me;; not){
   1 - out & out ;lie;{
   char lotte, my= dear,
   **let= !!me *!not+ ++die;
(char*)(lie=

"The gloves are OFF this time, I detest you, snot\n\0sed GEEK!");
   do {not= *lie++ & 0xF00L* !me;
   #define love (char*)lie -
   love 1s *!(not= atoi(let
   [get -me?
(char)lotte-

(char)lotte: my- *love -
   'I' - *love - 'U' -
   'I' - (long) - 4 - 'U' ])- !!
   (time =out= 'a'));} while( my - dear
   && 'I'-1l -get- 'a'); break;}}
(char)*lie++;
```

Nota Bene: Code should be readable… ;-)

(Source: signed char lotte)