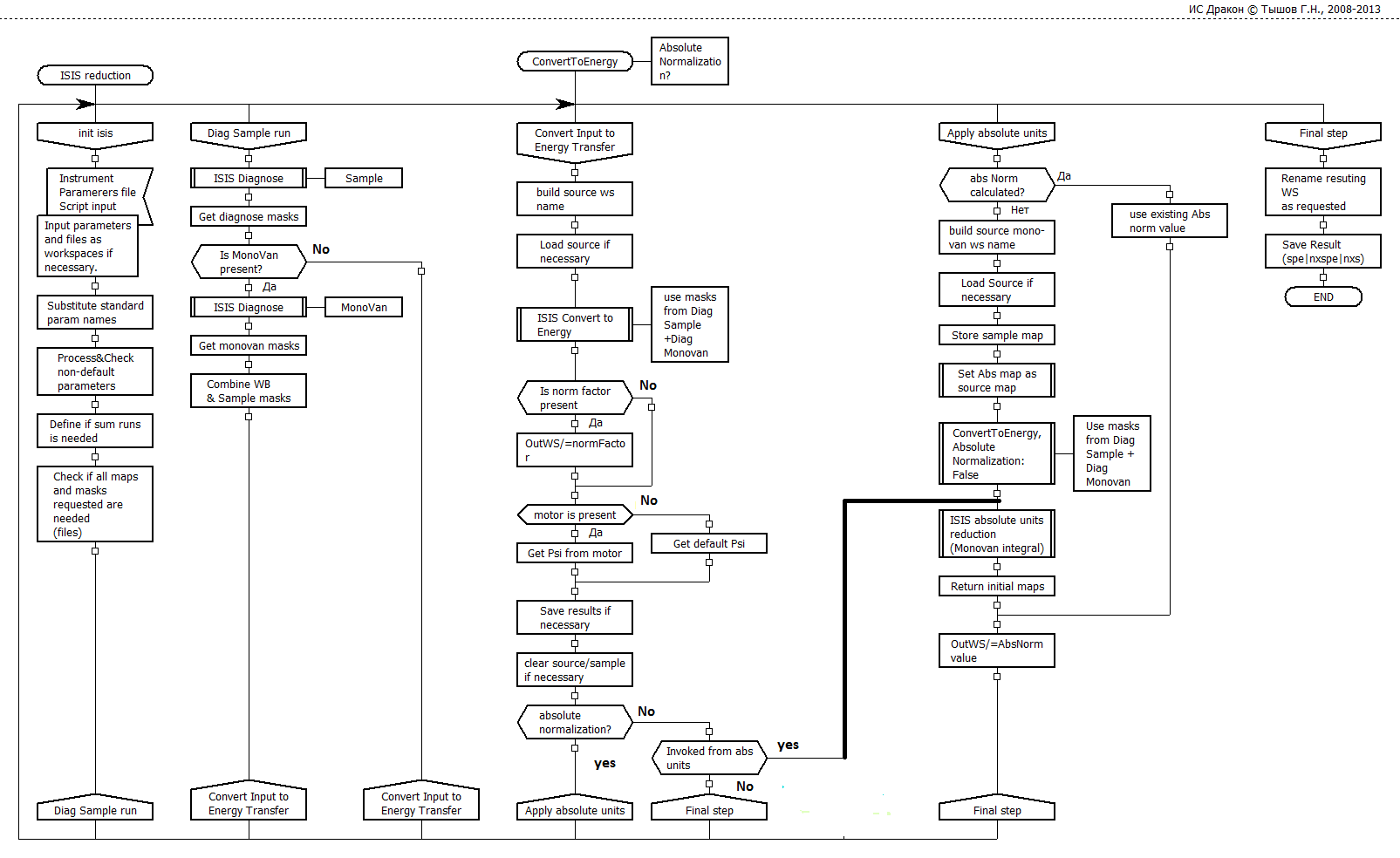
# SNS/ISIS reduction comparison

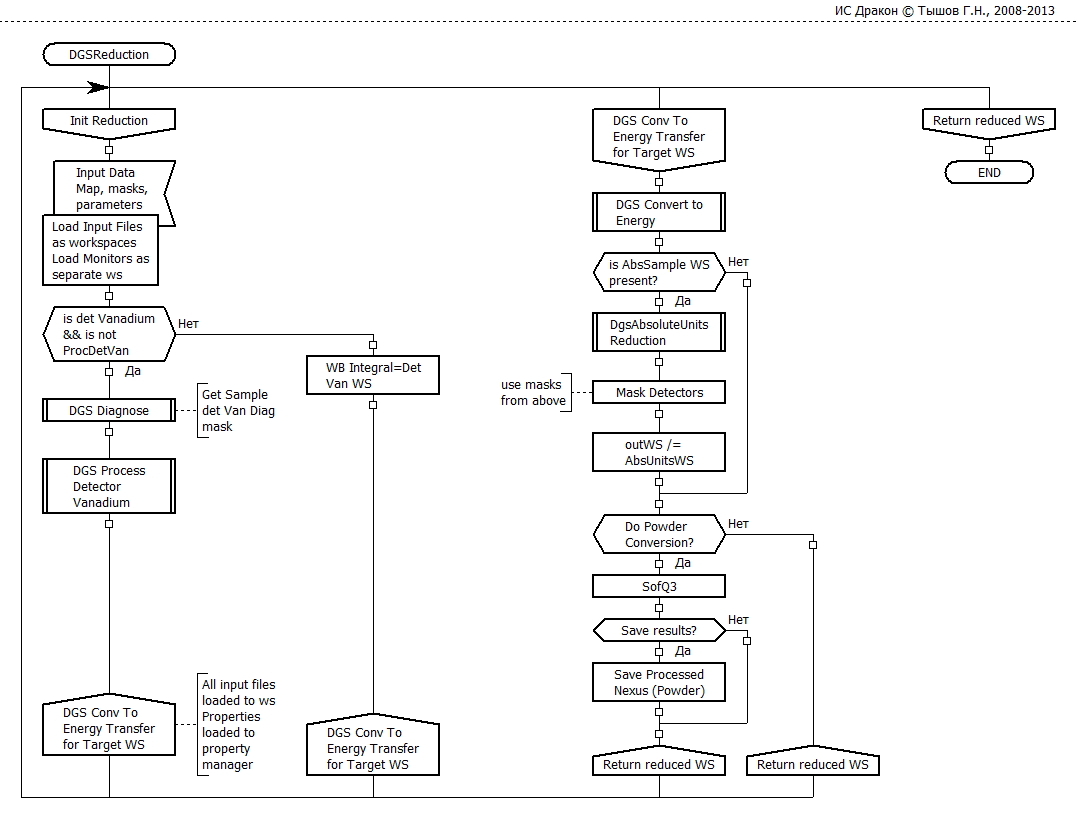
Used terminology:

|  |  |
| --- | --- |
| ISIS, python class methods and functions: | SNS, Algorithms |
| dgreduce(…) | DGSReduction(…) |
| Conversion from time-of-flight to energy transfer for direct geometry spectrometers. | |
| convert\_to\_energy(…) – class ConvertToEnergy | DgsConvertToEnergyTransfer(…) |
| Diagnose bad detectors and generate masks | |
| diagnose(…) -- module diagmose | DGSDiagnose(…) |
| Processing the detector vanadium in the form required for the sample data normalisation in the convert to energy transfer process. | |
| do\_white(…)– class ConvertToEnergy | DgsProcessDetectorVanadium(…) |
| Change Instruments detectors according to map/masks | |
| remap(…)– class ConvertToEnergy | DGSRemap(…) |
| Normalise data via a given incident beam parameter. For SNS, monitor workspaces need to be passed? (Current/Monitor/None) | |
| normalise(…)– class ConvertToEnergy | DGSPreprocessData |
| Take an absolute units sample and convert it to an integrated value for that sample. Corresponding detector vanadium can be used in conjunction with the data reduction. | |
| + diagnose + convert\_to\_energy +  get\_abs\_normalization\_factor() | DGSAbsUnitsReduction |

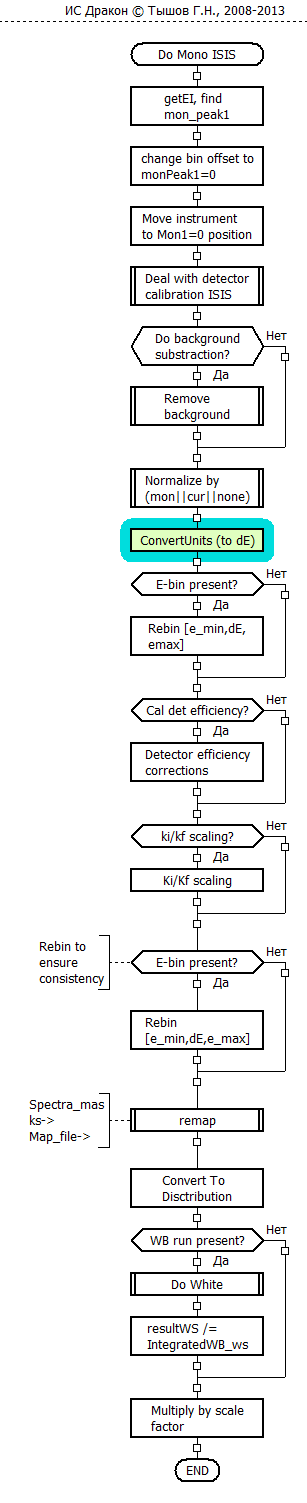


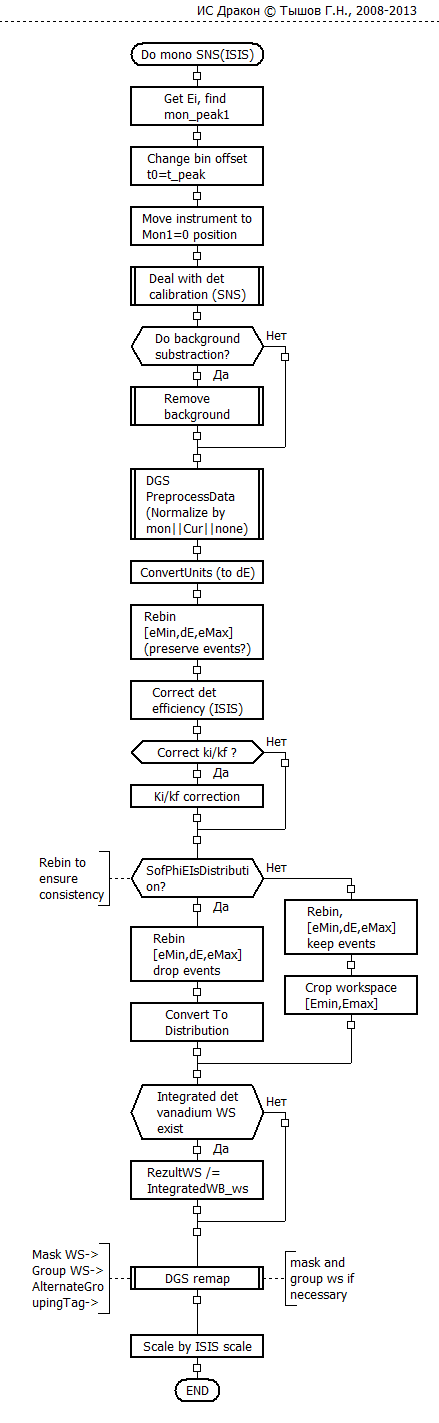
# ISIS Workflow

# SNS Workflow



# Convert To Energy





SNS only Conversion:

1) Can not be easy run for ISIS

2) T-zero

3) Different Detector efficiency correction procedure

4) Already No Det Calibration block

5) SNS Specific background removal mode

# Needs to be done

## SNS reduction for ISIS

ISIS remove background for events does not work?

Detectors corrections performed twice? At load and in the old place.

Monovanadium is not diagnosed?

What about default parameters. Where they came from (not sure, may be correctly – to clarify) ?

Energy independent cross-section?

Where is multirep mode?

General testing and verification.

## ISIS reduction for ISIS

Move detectors at loading – (almost done)

Separate monitor and run workspaces (must for event mode)

Background removal in Event mode

Better data & parameters separation?

Improvements for GUI (if use it)

# What would we have?

## Short Term:

# ISIS

Efforts to improve Current reduction (bugs) and introduce new features.

# SNS

Efforts to understand and verify current SNS for ISIS reduction

Efforts to fix some already identified problems.

Efforts to verify new features already there (e.g. event mode).

## Medium Term:

Code with poor data flow insulation, but clearer operational steps

Code which have a lot of legacy stuff which may not be well understood or not used any more.

GUI needs some improvement

Some places are not written very well

Better data flow separation but steps is much more difficult to understand.

Cleaner code written on the basis of rewriting of the old python code.

GUI OK (there is question about where defaults come)

Better written code though dodgy places can be still found.

## Long Term:

Python code – better wrapper language. Any IS expected to know it?

Easier and faster modifications. No problem to change something during the cycle

Code written and maintained by usC++ code. Limited number of people can modify and develop it

Difficult to change and modify Modifications within Mantid development cycle or under developer installation.

SNS develops and maintains the code?