GOCEXML2ASCII – an XML to ASCII converter for GOCE level 2 EGG_NOM and SST_PSO data

Matthias Roth

Institute of Geodesy, University of Stuttgart
matthias.roth@gis.uni-stuttgart.de

1. Motivation

ESA provides the GOCE data in XML format. This format is very easy to understand for a human reader. However, it has to be parsed by a computer to extract the data before it can be used in the computations. Usually, an XML file is parsed into a tree, which entails that the computer is equipped with sufficient memory to hold the complete tree structure.

The parsing is also time consuming, especially if it has to be done repeatedly in case of recomputations. Hence, in a normal workflow, the XML data will be converted to plain ASCII files (tables) in a preprocessing step prior to the actual computations. The offical GOCE L1b-L2 XML parser is available at ESA's website. It is programmed in Perl which is an interpreted programming language. Therefore, the conversion is slowed down by the interpretation step. It needs ~20 min for a single SST_PS0_2 file (a typical SST_PS0_2 file has a size of ~460 MB). The advantage of this parser is that it can be run on any operating system if Perl is installed.

The faster GOCEPARSER of the Finnish Geodetic Institute (Arsov, 2012) is implemented in C++. According to Arsov, the GOCEPARSER needs ~7 min to convert a SST_PS0_2 file which is the largest file type. Also the memory demand is much lower, because the XML file is parsed sequentially. The parser's size is around 8 MB and can only be run on a Windows operating system without modification. It is possible to convert several files at once, however they need to be unpacked from their archives in advance. Hence a big hard disk drive is needed.

Because our production system is running Ubuntu Linux and to overcome the above mentioned demands for a large disk capacity, we decided to implement our own parser.

2. The GOCEXML2ASCII parser

We decided to implement our parser in C using the XML2 library. The XML2 library provides functions for a very fast parsing of an XML document. Functions are available for either parsing the whole document to a tree structure in memory (memory consuming) or for reading the data sequentially. In the latter case, the user has to keep track of the actual depth and tags in the XML tree. However, we chose this method to keep the memory demand at its minimum.

In contrary to the above mentioned parsers, our parser can only convert EGG_NOM_2 and SST_PS0_2 at the moment. An extension to other GOCE data (e. g., level 1b) is easily possible. Our parser is very small (source code size ~60 kB, executable size ~30 kB) and fast (~2.5 min for a SST_PS0_2 file). We decided to implement also a small amount of date calculations into the parser. Hence, the parser already converts the date of a satellite position from the given format (year, month, day of month, hour, minute, second) to GPS seconds. This simplifies and also speeds-up further data conversions.

Usage:

```
./gocexml2ascii --if=<file.DBL> --hdr=<file.HDR> \
      --dir=<I/O directory>
```

It is not allowed to use a pathname for the `--if` and `--hdr` switches. The XML files have to reside in the directory specified by `--dir` (a "/" is mandatory after the directory name). The ASCII output will be written to the same directory.

3. Associated bash shell skripts and MATLAB function

The parser itself works only on a single file. To convert several files at once, it is possible to wrap a shell skript around the parser call, to feed the parser with the GOCE XML files.

Another goal of the shell skripting was, to keep the GOCE data compressed. E. g., all 424 available EGG_NOM_2 files (September 1012) have a compressed size of ~11 GB, which is blown up in uncompressed state to ~100 GB. The idea behind our shell skript is, to unpack one file to a temporary directory, convert the XML data into ASCII format, compress the ASCII data into a new archive, delete the contents of the temporary directory and proceed with the next file.

We decided to use the `zip`-archives for the output, because MATLAB contains routines for unpacking such archives. The reason to use `7z` as the packer is that it produces even smaller `zip`-archives than the standard packer for this format. For further conversions and calculations in MATLAB, we apply the same concept of first unpacking the data to a temporary directory, reading the data and deleting the folder afterwards.

Bash shell skript for EGG_NOM_2 conversion:

```
#!/bin/bash

for i in ./EGG_NOM_2/*.TGZ ; do
  temp=./tempEGG ;
  ascii=./EGG_NOM_2_ascii ;

  echo creating directory "${temp}"
  if [ -e ${temp} ] ;
  then
    echo ... already exists ... deleting its content.
    rm -r ${temp}
    mkdir ${temp}
  else
    mkdir ${temp}
  fi

  echo creating directory "${ascii}" ;
  if [ -e ${ascii} ] ;
  then
    echo ... already exists ... moving on.
  else
     mkdir ${ascii}
  fi

  file=${i:12} ;
  pos=‘expr index "$file" .‘ ;
  file=${file:0:$pos} ;

  echo working on $file ;

  echo extracting $i to $temp ;
  tar -zxvf $i -C $temp ;

  echo transforming ${temp}/${file} ... ;
  ./gocexml2ascii/gocexml2ascii --if=${file}DBL --hdr=${file}HDR \
      --dir=${temp}/

  echo packing to ${ascii}/${file}zip
  7z a -tzip ${ascii}/${file}zip ${temp}/*.ascii ${temp}/*.pdf

  mv ${i} ${i}.finish ;

  echo deleting ${temp}
  rm -r ${temp}
done
```

MATLAB function for unpacking zipped ASCII data:

```
unzip(<file.zip>, <directory>)
```

4. Source code of parser and shell skripts

The source code of the parser, as well as the bash shell script collection, are available on request. Please contact the author for it.

Acknowledgement

This work was supported by the European Space Agency (ESA) within the project UWB/GOCE-GDC – ITT AO/1-6367/10/NL/AF, STSE-GOCE+, Theme 2.

References

Arsov K (2012), "GOCEPARSER – A program to parse GOCE level 1b and level 2 data", EGU 2012, Vienna, Poster.