Testing of the COPE framework for processing observations

Mohamed Anis Satouri, December 2016
National Institute of Meteorology, Tunisia
anissatouri@gmail.com

In collaboration with Alena Trojakova
Czech Hydrometeorological Institute

Table of Contents

1. INTRODCTION

The Continuous Observation Processing Environment (COPE) project was initiated by ECMWF and is being developed in a collaboration with Météo-France, HIRLAM and ALADIN/LACE. It was proposed because the current observation processing chain including quality control which could be found in several places (e.g. pre-processing, external pre-screening, screening within assimilation...) is not optimal. The COPE framework will replace the packages OULAN / BATOR (and BUFR2ODB at ECMWF), to improve the pre-processing of observations for use in NWP.
The main objectives of COPE are to:

- ➢ Design the concepts for a scalable (quasi-) continuous observation monitoring and processing framework;
- ➢ Build the necessary infrastructure components and modify any existing systems (if required) to enable the implementation.

The aim of this stay is to test the COPE framework to reproduce the current observation pre-processing chain with the available COPE prototypes. The idea was to compare the COPE's ODB with Oulan/Bator's ODB.

The current design of the pre-processing chain for synop observations is as follow:

SYNOP TAC/BUFR format	**"decoding"**	local database (BUFR/NETCDF...)	OULAN	OBSOUL ASCII	BATOR	ODB-1

The COPE data flow for synop observations is as follow:

SYNOP BUFR format	B2O (bufr2ODB)	ODB-2	COPE FILTERS	**ODB-2** (filtered)	odb2_to_odb1	ODB-1

- ODB-1 is a column based memory representation used inside the ARPEGE/IFS (efficient for memory and parallelization), whereas ODB-2 is a line-based disk representation designed for archiving of ODB (efficient for disc access).
- OULAN / BATOR programs has been developed by Météo France and are used for conversion of observations to ODB-1 to be used in data assimilation applications within ARPEGE/ALADIN. BATOR has been extended to handle BUFR data directly, so the decoding to an internal database and OULAN processing is becoming obsolete.
- B2o (bufr2ODB) converts BUFR files to ODB-2 format, this library has been distributed as part of the IFS source code.
- COPE filters are a sequence of transformations (quality checks, unit conversions, computation of derived parameters e.g. wind speed and direction to u and v components, height to pressure coordinates, dumb thinning, bias correction, blacklisting) applied to each observation. Output from filters are written in different ODBs (one ODB per observation group but we could increase the granularity (by reportype) to increase parallelism).
- The conversion from ODB-2 data to ODB-1 database is done through ODB-API by the tool odb2_to_odb1.x.

2. TESTING OF THE COPE FRAMEWORK

Testing of the COPE framework comprised an installation of all libraries, testing and validation using the model configuration for an analysis using the optimal interpolation method (CANARI).

1. Compiling COPE and COPE related software:

Many thanks to Eoin Whelan for summarizing the instructions on how all relevant packages were built, and for his valuable help. A step-by-step set of instructions on how to compile COPE and COPE related software could be found in the following link:
https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/ObservObservationPreproc/Cope

The COPE installation requires the following:

- "Support" software packages are odb_api_bundle (0.15.2) and libemos (4.4.2)
  odb_api_bundle (0.15.2) include: ecBuild (2.4.0) eckit (0.14.0) and metkit (0.3.0).
- “Main" software packages are : odb (40t1.01), ODB API (0.15.4), b2o (40t1.01) and COPE.

The compilation of the libraries ecBuild (2.4.0), eckit (0.14.0) and metkit (0.3.0) was smooth. For libemos (4.4.2) we had firstly to install fftw-3.3.5 and we had to use static libraries. For odb (40t1.01) we noticed that we should use a newer version of GCC (error: unrecognized command line option "-fopenmp"), and a newer version of python, so we recompiled the libraries with (GCC) 4.8.5, and we are using now Python 2.7.3. The libraries were installed, but the 2 tests of make check are failing. For ODB API (0.15.4) we had to disable netcdf and to use a recent version of flex, actually the used one is flex 2.6.3. (86% tests passed, 33 tests failed out of 241 (by checking Testing/Temporary/LastTest.log, we figured out that we had to download some data manually to resolve this)). For b2o (40t1.01), 50% of the make check tests were successful (the conversion ones), and all the comparison tests failed. Finally the compilation of COPE was smooth.
NB. The detailed compilation process is included in appendix.

2. Testing of COPE libraries:

The testing and validation was complicated by the fact that the COPE framework was available within the harmonie_40h1_cope branch, while only cy38t1 was available at CHMI. A single synoptic observation in BUFR format (BUFR SYNOP) available from GTS was used as starting point. The second test was done with a sample provided by Eoin Whelan with conventional BUFR data from ECMWF over Scandinavia (Harmonie testbed test domain 50 x 50).

The GTS synop bufr file available on CHMI server yaga:

```
yaga:/work/mma241/cope/b2o_data/ISMD01_OKPR_100000_547
```

was used to produce ODB-1 database via following sequence **b2o** → **cope filters** → **odb2_to_odb1.**

- The **b2o** is used to obtain an ODB-2 data from the bufr file as follow: (how to execute b2o):

```
b2o ISMD01_OKPR_100000_547 -o ISMD01_OKPR_100000_547.odb2
```

Doing so we ended on this crash:

```
Error: No obstype/codetype mapping for bufrtype=0, subtype=0, all-sky=F
 ABORT! find_obstype_codetype
```

This error seems to be related to the mapping file (it doesn't include this configuration) To be 100% sure that this has no relation with our b2o executable we tested one bufr file included in the b2o test directory:

```
b2o synop-land-manual.input -o synop-land-manual.odb2
```

and effectively it was running well and it produces an ODB-2 file.

Investigating the above error:

The subroutine find_obstype_codetype in the routine odbmap_reportype.F90: searches for matching code combinations from config file mappings. To every bufrtype and subtype correspond an obstype and codetype.

bufrtype = ksec1(6): Bufr message type (Bufr Table A)
subtype = ksec1(7) : Bufr message subtype (local use)

For the synop bufr file included in the test package (synop-land-manual.input which is a bufr edition 3) we have:

bufrtype = 0: Surface data–land
subtype = 1: Intermediate synoptic observations from fixed-landstations (SYNOP)

The corresponding obstype and codetype in the config file mappings are:

obstype = 1: Land SYNOP and SHIP reports
codetype = 11: SYNOP land manual

For our synop testing file ISMD01_OKPR_100000_547 (which is a bufr file edition 4) we have:

bufrtype = 0: Surface data–land
subtype = 0: Hourly synoptic observations from fixed-land stations(SYNOP)

This configuration is not included in the config file mappings, that's for why b2o was crashing.
As far as I understood the corresponding obstype/codetype are also obstype=1 and codetype=11 so I added this line to the config file mappings :

```
/home/mma/mma241/metapp/b2o/40t1.01/gnu/share/b2o/odb_code_mappings.dat
16000  , 17  , 0  , 0  , 1  , 11  , -1  , -1  , -1  , -1  ,
```

So b2o runs without crashing and we obtained an ODB-2 output : ISMD01_OKPR_100000_547.odb2
I'm wondering if ECMWF is not using the hourly synoptic observations so they use only the intermediate ones! This is definitely illogical, because this will negatively influence the 4-dvar system. Otherwise they are encoding these observations locally in a specific way which seems to be more logical.

Some hints :

```
# input BUFR file could be inspected using b2o :
b2o -q 'select lat,lon,statid,obsvalue,varno' ISMD01_OKPR_100000_547
# the content of an ODB-2 file could be checked using ODB-API, e.g.
odbsql 'select lat,lon,statid,obsvalue,varno' -i ISMD01_OKPR_100000_547.odb2
```

- The **Filters** are applied by 'Cope'. It carries out QC/filtering based on contents of *.json files. It contains a macro language and the real content is in C sources, e.g. /home/mma/mma241/test_ecSource/cope/src/cope/filters/LocationValidator.cc

For example this is what looks like synop.json file

```
{
   "filters": [
      { "name": "LocationValidator" },
      { "name": "DateTimeValidator" },
      { "name": "InstrumentTypeAssigner" },
      { "name": "MfVerticalCoordinateAssigner" },
      { "name": "LandSynopVerticalCoordinateAssigner" },
      { "name": "BiasCorrector" },
      { "name": "WindComponentsAssigner" },
      { "name": "PrescribedErrorAssigner",
          "options": {
              "statistics_file": "error_statistics.csv"
          }
      },
      {
        "name": "RelativeHumidityAssigner",
        "options": {
             "apply_Td_gt_T_check": false,
             "svp": "buck"
          }
      },
      { "name": "PrecipitationSplitter" },
      { "name": "FinalErrorAssigner" },
      { "name": "DegreesToRadiansConverter", "disabled": false },
      { "name": "FinalChecker", "disabled": true }
   ]
}
```

Follow how to execute cope :

```
cope screen -j synop.json -s conv.schema.sql -i ISMD01_OKPR_100000_547_1.odb2 -o 
ISMD01_OKPR_100000_547_1.odb2_filetered
```

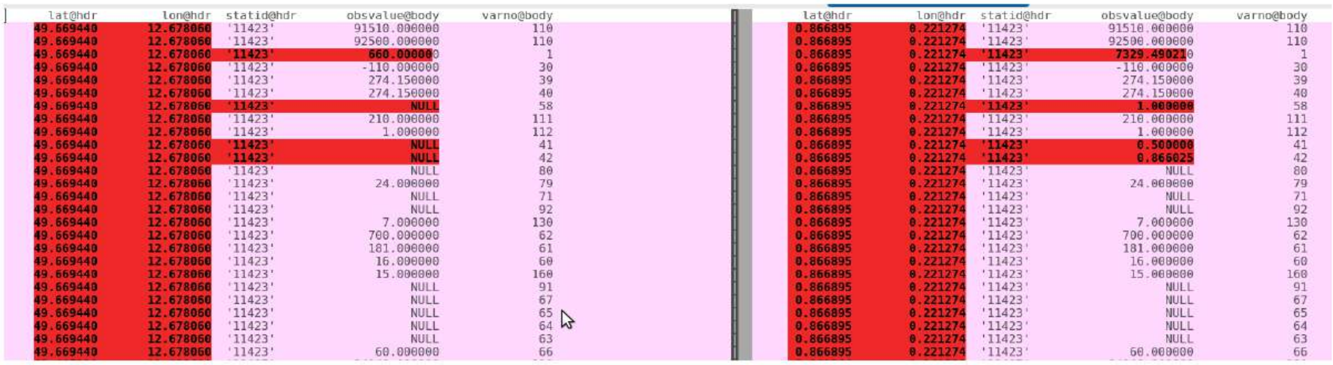
I tried to compare the two ODB-2 files (with and without filters)

```
odb compare ISMD01_OKPR_100000_547.odb2  ISMD01_OKPR_100000_547.odb2_filtered
```

but I got this error : Assertion failed: "Number of columns must be the same"

Comparison of the outputs of odbsql select for the two odb2 files was used for comparison:

```
odbsql 'select lat,lon,statid,obsvalue,varno' -i  ISMD01_OKPR_100000_547.odb2 > log_odb2 2>&1
odbsql 'select lat,lon,statid,obsvalue,varno' -i ISMD01_OKPR_100000_547.odb2_filtered > 
log_odb2_filtered 2>&1
gvimdiff  log_odb2 log_odb2_filtered
```



```
   lat@hdr      lon@hdr  statid@hdr    obsvalue@body   varno@body
 49.669440    12.678060  '11423'     91510.000000          110
 49.669440    12.678060  '11423'     92500.000000          110
 49.669440    12.678060  '11423'       660.000000            1
 49.669440    12.678060  '11423'      -110.000000           30
 49.669440    12.678060  '11423'       274.150000           39
 49.669440    12.678060  '11423'       274.150000           40
 49.669440    12.678060  '11423'             NULL           58
 49.669440    12.678060  '11423'       210.000000          111
 49.669440    12.678060  '11423'         1.000000          112
 49.669440    12.678060  '11423'             NULL           41
 49.669440    12.678060  '11423'             NULL           42
 49.669440    12.678060  '11423'             NULL           80
 49.669440    12.678060  '11423'        24.000000           79
 49.669440    12.678060  '11423'             NULL           71
 49.669440    12.678060  '11423'             NULL           92
 49.669440    12.678060  '11423'         7.000000          130
 49.669440    12.678060  '11423'       700.000000           62
 49.669440    12.678060  '11423'       181.000000           61
 49.669440    12.678060  '11423'        16.000000           60
 49.669440    12.678060  '11423'        15.000000          160
 49.669440    12.678060  '11423'             NULL           91
 49.669440    12.678060  '11423'             NULL           67
 49.669440    12.678060  '11423'             NULL           65
 49.669440    12.678060  '11423'             NULL           64
 49.669440    12.678060  '11423'             NULL           63
 49.669440    12.678060  '11423'        60.000000           66
```

```
   lat@hdr      lon@hdr  statid@hdr    obsvalue@body   varno@body
  0.866895     0.221274  '11423'     91510.000000          110
  0.866895     0.221274  '11423'     92500.000000          110
  0.866895     0.221274  '11423'      7329.490210            1
  0.866895     0.221274  '11423'      -110.000000           30
  0.866895     0.221274  '11423'       274.150000           39
  0.866895     0.221274  '11423'       274.150000           40
  0.866895     0.221274  '11423'         1.000000           58
  0.866895     0.221274  '11423'       210.000000          111
  0.866895     0.221274  '11423'         1.000000          112
  0.866895     0.221274  '11423'         0.500000           41
  0.866895     0.221274  '11423'         0.866025           42
  0.866895     0.221274  '11423'             NULL           80
  0.866895     0.221274  '11423'        24.000000           79
  0.866895     0.221274  '11423'             NULL           71
  0.866895     0.221274  '11423'             NULL           92
  0.866895     0.221274  '11423'         7.000000          130
  0.866895     0.221274  '11423'       700.000000           62
  0.866895     0.221274  '11423'       181.000000           61
  0.866895     0.221274  '11423'        16.000000           60
  0.866895     0.221274  '11423'        15.000000          160
  0.866895     0.221274  '11423'             NULL           91
  0.866895     0.221274  '11423'             NULL           67
  0.866895     0.221274  '11423'             NULL           65
  0.866895     0.221274  '11423'             NULL           64
  0.866895     0.221274  '11423'             NULL           63
  0.866895     0.221274  '11423'        60.000000           66
```

fig1. Comparison between the 2 odbsql select log, the original at left and the filtered one at right

The filters applied by “Cope” transformed latitude and longitude from degrees to radians. Relative humidity is computed from temperature and dew point (varno = 58 => 2m relative humidity). And wind u/v components are computed from speed and direction (varno = 41 => 10m meridional component ; varno=42 => 10m zonal component).

- The **odb2_to_odb1.x** is used to convert the ODB-2 file to an ODB-1 containing npools pools.

Basically this is the command how to execute odb2_to_odb1

```
odb2_to_odb1.x -i $base -t groupid17.tables -o ECMA -npools $npools
```

The groupid17.tables file defines the ODB1 schema. "17" refers to conventional data. See http://apps.ecmwf.int/odbgov/group/ for the full list. A script with more details could be found in the following path: yaga: /work/mma241/odb2_to_odb1/test/test/run.ksh
This script uses odb ecml to split ODB-2 file and odb2_to_odb1.x to create ODB-1

Running the script whith -e option exits whithout creating an ODB-1 for npools > 1 on this error

```
While comparing rows number 105, columns 0 found different.
Values different in column seqno@hdr: 5 is not equal 3
data1[0] = 5.000000e+00
data2[0] = 3.000000e+00
md1[0] = name: seqno@hdr, type: INTEGER, codec: int8, range=<1.000000,7.000000>
md2[0] = name: seqno@hdr, type: INTEGER, codec: int8, range=<3.000000,4.000000>
Exception: Values different in column subsetno@hdr: 5 is not equal 3
```

set -xe / npools = 4	Ending on the above error
set -xe / npools = 1	Produces an ODB-1 with 1 pool
set -x / npools = 4	Produces an ODB-1 with 4 pools

A comment from Eoin was “my guess is that your odb2_to_odb1 problems are related to how your ODB1 schema is defined”.

3. Running BATOR & CANARI

The final step of the validation is to use created ODB-1 in the CANARI configuration. The ODB-1 (ECMA) produced by the COPE framework for cycle 40 was tested in Toulouse through CANARI, but the configuration failed to handle it (same test was done at CHMI for cy38t1 and with the same error).

The script to run canari on beaufix could be found in the following path :
beaufix : /home/gmap/mrpe/satouria/canari_cope/canari_cope_anis

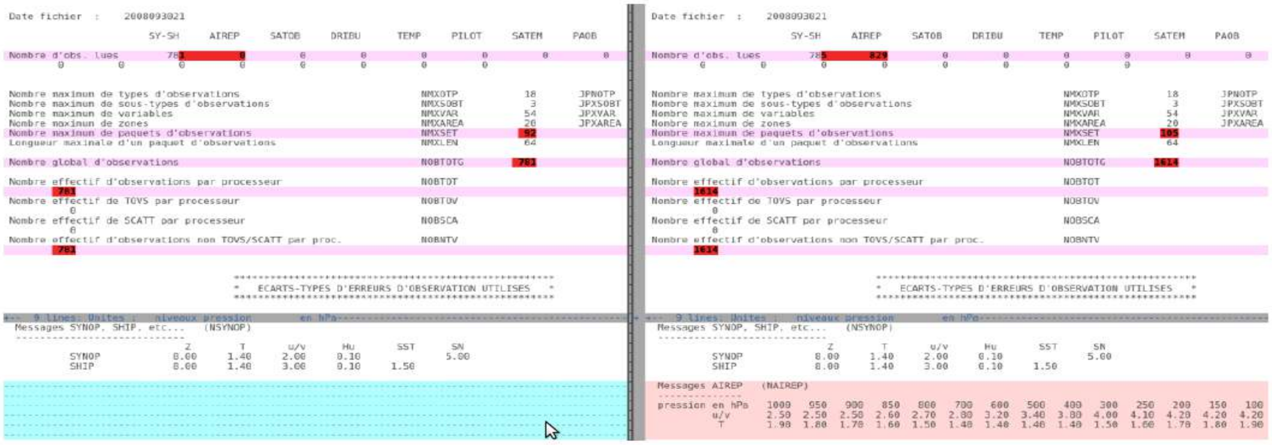
The error message follows:

```
READOBA: Opened ODB-database="ECMA with mode="OLD" : npools & $NMXUPD are
         1          3  LECFIO= T
 ADDVIEWDB("time_numtsl" : db="ECMA") : total#, dbhandle, viewhandle,
thread-id =    1    1  1124618241   1
forrtl: severe (174): SIGSEGV, segmentation fault occurred
Image              PC                Routine Line        Source
MASTERODB         0000000001164717  readoba_ 231  readoba.F90
MASTERODB         00000000019A5012  obadat_ 115  obadat.F90
MASTERODB         000000000117F155  sudimo_ 138  sudimo.F90
MASTERODB         0000000000949D67  su0yomb_ 412  su0yomb.F90
MASTERODB         00000000005FDEAE  cnt0_ 134  cnt0.F90
MASTERODB         00000000005FDD87  MAIN__ 76  master.F90
```

In order to identify this issue the sample of data from Harmonie testbed (Scandinavian domain 50 x 50) was used. This data contains conventional BUFR data from ECMWF. Investigations pointed a need for a fix in the routine shuffle_odb.F90 available in 40h1 HARMONIE branch.

The fix consists on adding a new key LL_ecmwf = .TRUE.
The default and updated routines can be found on beaufix:
~trojakova/pack/cy40t1_bf05.01.IMPI500IFC1310.2x.pack/src/main/odb/cma2odb/shuffle_odb.F90
~satouria/pack/cy40t1_test_cope/src/local/odb/cma2odb/shuffle_odb.F90

The following figure shows a comparison between the listings of CANARI 40t1 using data from Harmonie testbed provided by our HIRLAM colleague Eoin Whelan vs data filtered through COPE framework on yaga
cd /home/gmap/mrpe/satouria/canari_cope
gvimdiff canari_cope_anis.o5328335 canari_cope_anis.o5403100
NB: Again it seems that we have problem mainly with odb2_to_odb1 step, which should be investigated deeply. This was the last task done during this stay.



```
Date fichier  :    2008093021

                        SY-SH      AIREP      SATOB      DRIBU      TEMP      PILOT      SATEM      PAOB

Nombre d'obs. lues         781          0          0          0          0          0          0          0
        0          0          0          0          0          0          0          0

Nombre maximun de types d'observations                                       NMXOTP          18       JPNOTP
Nombre maximun de sous-types d'observations                                  NMXSOBT          3       JPXSOBT
Nombre maximun de variables                                                  NMXVAR          54       JPXVAR
Nombre maximun de zones                                                      NMXAREA         20       JPXAREA
Nombre maximun de paquets d'observations                                     NMXSET          92
Longueur maximale d'un paquet d'observations                                 NMXLEN          64

Nombre global d'observations                                                 NOBTOTG        781

Nombre effectif d'observations par processeur                                NOBTOT
       781
Nombre effectif de TOVS par processeur                                       NOBTOV
         0
Nombre effectif de SCATT par processeur                                      NOBSCA
         0
Nombre effectif d'observations non TOVS/SCATT par proc.                      NOBNTV
       781

                                            ****************************************************
                                            *    ECARTS-TYPES D'ERREURS D'OBSERVATION UTILISES   *
                                            ****************************************************

+--  9 lines: Unites :    niveaux pression        en hPa-------------------------------------------
 Messages SYNOP, SHIP, etc...    (NSYNOP)
 ----------------------------
                         Z          T        u/v         Hu        SST         SN
     SYNOP            8.00       1.40       2.00       0.10                  5.00
     SHIP             8.00       1.40       3.00       0.10       1.50
```

```
Date fichier  :    2008093021

                        SY-SH      AIREP      SATOB      DRIBU      TEMP      PILOT      SATEM      PAOB

Nombre d'obs. lues         785        829          0          0          0          0          0          0
        0          0          0          0          0          0          0          0

Nombre maximum de types d'observations                                       NMXOTP          18       JPNOTP
Nombre maximum de sous-types d'observations                                  NMXSOBT          3       JPXSOBT
Nombre maximum de variables                                                  NMXVAR          54       JPXVAR
Nombre maximum de zones                                                      NMXAREA         20       JPXAREA
Nombre maximum de paquets d'observations                                     NMXSET         105
Longueur maximale d'un paquet d'observations                                 NMXLEN          64

Nombre global d'observations                                                 NOBTOTG       1614

Nombre effectif d'observations par processeur                                NOBTOT
      1614
Nombre effectif de TOVS par processeur                                       NOBTOV
         0
Nombre effectif de SCATT par processeur                                      NOBSCA
         0
Nombre effectif d'observations non TOVS/SCATT par proc.                      NOBNTV
      1614

                                            ****************************************************
                                            *    ECARTS-TYPES D'ERREURS D'OBSERVATION UTILISES   *
                                            ****************************************************

+--  9 lines: Unites :    niveaux pression        en hPa-------------------------------------------
 Messages SYNOP, SHIP, etc...    (NSYNOP)
 ----------------------------
                         Z          T        u/v         Hu        SST         SN
     SYNOP            8.00       1.40       2.00       0.10                  5.00
     SHIP             8.00       1.40       3.00       0.10       1.50

 Messages AIREP   (NAIREP)
 --------------
 pression en hPa   1000   950   900   850   800   700   600   500   400   300   250   200   150   100
       u/v         2.50  2.50  2.50  2.60  2.70  2.80  3.20  3.40  3.80  4.00  4.10  4.20  4.20  4.20
        T          1.90  1.80  1.70  1.60  1.50  1.40  1.40  1.40  1.40  1.50  1.60  1.70  1.80  1.90
```

Fig2. comparison between the listings of canari 40t1, odb merge (Eoin’s data) at left vs odb filtered through cope on yaga at right

SUMMARY

The goal of this stay was to get more familiar with the COPE framework and to reproduce the current observation pre-processing chain. The COPE prototypes were succesfully installed and they seems to work TECHNICALLY. Several issues have to be still investigated, in particular

- b2o config file mapping setting for a processing of GTS BUFR data;
- cope filters requires deeper understanding and cross-validation with respect to BATOR;
- odb2_to_odb1 use for npool > 1 still does not work;
- shuffle_odb.F90 modification have to be better understood.

Overall detailed validation is essential for a progress on use of the COPE.

Furthermore, I'm wondering to what point the COPE framework will be efficient and/or beneficial for LAM applications in comparison with BATOR, especially with respect to it’s maintenance and the maintenance related software. Other open question is how COPE will deal for example with a complement of a GTS message or how is it going to handle duplicates, corrections (for erroneous messages), late arrivals observations reports etc.

Acknowledgement

At the end of this report I want to thank Eoin Whelan for providing valuable support to this work, Alena Trojakova for her valuable guidance, advices and entire support during my stay, as well many thanks to the entire NWP team for their warm welcome and hospitality.

Here are some links that may be of interest:

- https://software.ecmwf.int/wiki/display/COPE/COPE: COPE wiki (restricted access)
- http://www.rclace.eu/File/Data_Assimilation/2014/201406_COPE_Reading_report.pdf: Report on the COPE technical meeting, *Alena Trojáková*. ECMWF, Reading 9-12, June 2014
- http://www.cnrm.meteo.fr/aladin/IMG/pdf/copeovervieweoinwhelan.pdf: Overview of COPE, *Eoin Whelan*. Joint 24th ALADIN Workshop & HIRLAM All Staff Meeting 2014, 7-11 April 2014, Romania.
- https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/ObservationPreprocessing/Cope
- https://software.ecmwf.int/wiki/display/ODBAPI/Command+line+tools
- https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home
- https://software.ecmwf.int/wiki/display/ECC/BUFR+tools
- https://software.ecmwf.int/wiki/display/EMOS/Installation+Guide

APPENDIX

A. The detailed compilation process COPE and COPE related software:

The needed packages could be found on yaga under this path:
/home/mma/mma153/work/test_ecmwf_releases
Compiling the different packages following instructions from Eoin, see :
https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/ObservationPreprocessing/Cope

Preparation :

```
mkdir -p $HOME/test_ecmwf_releases
mkdir -p $HOME/test_ecSource
cp odb_api_bundle-0.15.2-Source.tar.gz $HOME/test_ecmwf_releases/
cp libemos-4.4.2-Source.tar.gz $HOME/test_ecmwf_releases/
cd $HOME/test_ecmwf_releases
gunzip odb_api_bundle-0.15.2-Source.tar.gz
tar -xvf odb_api_bundle-0.15.2-Source.tar
gunzip libemos-4.4.2-Source.tar.gz
tar -xvf libemos-4.4.2-Source.tar
```

ecBuild (2.4.0)

```
cd $HOME/test_ecmwf_releases/odb_api_bundle-0.15.2-Source/ecbuild
mkdir build
cd build/
cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/metapp/ecbuild/2.4.0/gnu/
-DCMAKE_C_COMPILER=/soft/gfortran/bin/gcc
-DCMAKE_CXX_COMPILER=/soft/gfortran/bin/g++
-DCMAKE_Fortran_COMPILER=/soft/gfortran/bin/gfortran
 make
 make check
 make install
```

eckit (0.14.0)

```
cd $HOME/test_ecmwf_releases/odb_api_bundle-0.15.2-Source/eckit
mkdir build
cd build/
cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/metapp/eckit/0.14.0/gnu/
-DCMAKE_MODULE_PATH=$HOME/metapp/ecbuild/2.4.0/gnu/share/ecbuild/cmake
-DCMAKE_C_COMPILER=/soft/gfortran/bin/gcc
-DCMAKE_CXX_COMPILER=/soft/gfortran/bin/g++
-DCMAKE_Fortran_COMPILER=/soft/gfortran/bin/gfortran
 make -j 4
 make check
 make install
```

metkit (0.3.0)

```
cd $HOME/test_ecmwf_releases/odb_api_bundle-0.15.2-Source/metkit
mkdir build
```

```
cd build/
cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/metapp/metkit/0.3.0/gnu/
-DCMAKE_MODULE_PATH=$HOME/metapp/ecbuild/2.4.0/gnu/share/ecbuild/cmake/
-DECKIT_PATH=$HOME/metapp/eckit/0.14.0/gnu
-DCMAKE_C_COMPILER=/soft/gfortran/bin/gcc
-DCMAKE_CXX_COMPILER=/soft/gfortran/bin/g++
-DCMAKE_Fortran_COMPILER=/soft/gfortran/bin/gfortran
make -j 4
make check
make install
```

grib_api (1.18.0)

```
tar xvfz  /home/mma/mma153/work/test_ecmwf_releases/grib_api-1.18.0-Source.tar.gz
cd grib_api-1.18.0-Source
mkdir build && cd build
cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/metapp/grib_api
make
tar xvfz  /home/mma/mma153/work/test_ecmwf_releases/grib_api_test_data.tar.gz
ctest
make install
```

fftw (3.3.5)

```
tar xvf  /home/mma/mma153/work/test_ecmwf_releases/fftw-3.3.5.tar.gz
cd fftw-3.3.5
./configure --prefix=$HOME/metapp/fftw
make
make install
```

see https://software.ecmwf.int/wiki/display/EMOS/Installation+Guide

libemos (4.4.2)

```
cd $HOME/test_ecmwf_releases/libemos-4.4.2-Source
mkdir build
cd build/
cmake ..  -DCMAKE_INSTALL_PREFIX=$HOME/metapp/libemos/4.4.2/gnu
-DGRIB_API_PATH=$HOME/metapp/grib_api -DFFTW_PATH=$HOME/metapp/fftw
-DFFTW_LIB=$HOME/metapp/fftw/lib/ -DFFTW_USE_STATIC_LIBS=ON
-DCMAKE_C_COMPILER=/soft/gfortran/bin/gcc
-DCMAKE_CXX_COMPILER=/soft/gfortran/bin/g++
-DCMAKE_Fortran_COMPILER=/soft/gfortran/bin/gfortran
```

ERROR:

```
[ 31%] Building C object sandbox/CMakeFiles/emos_tool.dir/emos_tool.c.o
Linking C executable ../bin/emos_tool
/usr/bin/ld: Dwarf Error: found dwarf version '4', this reader only handles version 2 information.
../lib/libemosR64.a(jsymgg.F.o): In function `jsymgg_':
jsymgg.F:(.text+0x8d2): undefined reference to `dfftw_plan_many_dft_c2r_'
jsymgg.F:(.text+0x8dc): undefined reference to `dfftw_execute_'
jsymgg.F:(.text+0x8e6): undefined reference to `dfftw_destroy_plan_'
collect2: error: ld returned 1 exit status
make[2]: *** [bin/emos_tool] Error 1
```

```
make[1]: *** [sandbox/CMakeFiles/emos_tool.dir/all] Error 2
make: *** [all] Error 2
```

Correct way to link to the static library
=> -DFFTW_LIB=$HOME/metapp/fftw-3.3.5/lib/libfftw3.a

```
cmake ..  -DCMAKE_INSTALL_PREFIX=$HOME/metapp/libemos/4.4.2/gnu
-DGRIB_API_PATH=$HOME/metapp/grib_api -DFFTW_PATH=$HOME/metapp/fftw
-DFFTW_LIB= =$HOME/metapp/fftw-3.3.5/lib/libfftw3.a -DFFTW_USE_STATIC_LIBS=ON
-DCMAKE_C_COMPILER=/soft/gfortran/bin/gcc
-DCMAKE_CXX_COMPILER=/soft/gfortran/bin/g++
-DCMAKE_Fortran_COMPILER=/soft/gfortran/bin/gfortran
 make
 make test
 make install
```

odb (40t1.01)

```
cd $HOME/test_ecSource
tar xvf  /home/mma/mma153/work/test_ecmwf_releases/odb-40t1.01-Source.tar
cd odb-40t1.01/
mkdir build
export PATH=/soft/python/bin/:$PATH
cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/metapp/odb/40t1.01/gnu/
-DCMAKE_MODULE_PATH=$HOME/metapp/ecbuild/2.4.0/gnu/share/ecbuild/cmake/
-DODB_SCHEMAS="ECMA;CCMA" -DCMAKE_C_COMPILER=/soft/gfortran/bin/gcc
-DCMAKE_CXX_COMPILER=/soft/gfortran/bin/g++
-DCMAKE_Fortran_COMPILER=/soft/gfortran/bin/gfortran
```

It was at this stage were we figured out that we have to use a newer version of python and gcc.

```
!!  File "/home/mma/mma241/test_ecmwf_releases/odb40t1.01/cmake/odb_generate_table_names.py",
line 4, in ?
    import argparse
```

```
make -j 8
```

```
+> export PATH=/soft/python/bin/:$PATH
```

```
## Error: cc1: error: unrecognized command line option "-fopenmp" => use higher version of gcc (e.g.
4.8.5)
```

```
Test project /home/mma/mma241/work/test_ecmwf_releases/odb-40t1.01/build
    Start 1: test_createdb
1/2 Test #1: test_createdb ....................***Failed    0.00 sec
    Start 2: test_populate
2/2 Test #2: test_populate ....................***Failed    0.01 sec

0% tests passed, 2 tests failed out of 2
```

ODB API (0.15.4)

```
cd $HOME/test_ecSource
tar xvf  /home/mma/mma153/work/test_ecmwf_releases/odb_api-0.15.4-Source.tar
cd odb_api-0.15.4/
mkdir build
cd build/
cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/metapp/odb_api/0.15.4/gnu/
-DCMAKE_MODULE_PATH=$HOME/metapp/ecbuild/2.4.0/gnu/share/ecbuild/cmake/
-DECKIT_PATH=$HOME/metapp/eckit/0.14.0/gnu/
-DMETKIT_PATH=$HOME/metapp//metkit/0.3.0/gnu -DENABLE_MIGRATOR=ON
-DODB_PATH=$HOME/metapp//odb/40t1.01/gnu -DENABLE_FORTRAN=ON
-DENABLE_PYTHON=ON -DENABLE_NETCDF=OFF
  make -j 8
  make VERBOSE=1
```

```
ERROR: [ 17%] [FLEX][odblib__sqly_scanner] Building scanner with flex 2.5.31
cd /home/mma/mma241/test_ecmwf_releases/odb_api-0.15.4/src/odb_api &&
/usr/bin/flex -d -o/home/mma/mma241/test_ecmwf_releases/odb_api-
0.15.4/build_gnu.4.8.5/src/odb_api/sqll.tmp.c
/home/mma/mma241/test_ecmwf_releases/odb_api-0.15.4/src/odb_api/sqll.l
/home/mma/mma241/test_ecmwf_releases/odb_api-0.15.4/src/odb_api/sqll.l:2: unrecognized %option:
extra-type
make[2]: *** [src/odb_api/sqll.tmp.c] Error 1
```

=> %option: extra-type supported from flex 2.5.34 ! Flex upgrade is needed.

```
   make -j 8
   make test
```

```
86% tests passed, 33 tests failed out of 241

Label Time Summary:
odb_api             =   7.93 sec
odb_api_ecml        =   0.78 sec
odb_api_fortran     =   0.12 sec
odb_api_migrator    =   0.05 sec
odb_api_python      =   1.38 sec

Total Test time (real) =   8.93 sec

The following tests FAILED:
         18 - get_odb_migrator_test_data (Failed)
         21 - python_odb_api_get_test_data (Failed)
         22 - test_python_odb_api.py (SEGFAULT)
         23 - legacy_test_python_odb_api.py (Failed)
         24 - get_migrator_test_data (Failed)
         25 - get_odb_api_test_data (Failed)
         26 - get_odb2_to_odb1_data (Failed)
         27 - test_odb2_to_odb1 (Failed)
         28 - get_odb2netcdf_data (Failed)
         29 - Test_AggregateFunctions (Failed)
         33 - Test_Bitfields (Failed)
         50 - Test_Decoding (Failed)
         51 - Test_DispatchingWriter (Failed)
         72 - Test_FastODA2Request (Failed)
         74 - Test_FastODA2Request3 (Failed)
        176 - Test_SelectDataHandle (Failed)
        177 - Test_SelectIterator (Failed)
        180 - Test_SelectStarAt (Failed)
        187 - Test_Star (Failed)
        199 - Test_bitfields_hash_operator (Failed)
        206 - Test_meta_data_reader_checks_if_file_truncated (Failed)
        207 - Test_meta_data_reader_fails_scanning_corrupted_file (Failed)
        229 - test_ec_archiving.ecml (Failed)
        230 - test_mo_archiving.ecml (Failed)
        231 - test_sql_splitting.ecml (Failed)
        232 - test_chunk.ecml (Failed)
        233 - test_chunk2.ecml (Failed)
        234 - test_embedded_ecml_in_from_clause.ecml (Failed)
        237 - test_create_partitions.ecml (Failed)
        238 - test_server_side_processing.ecml (Failed)
        239 - test_stage.ecml (Failed)
        240 - get_mars_client_test_data_mo (Failed)
        241 - get_mars_client_test_data_ec (Failed)
```

The above tests failed before we download the data manually.

b2o (40t1.01)

```
cd $HOME/test_ecmwf_releases
tar xvf  /home/mma/mma153/work/test_ecmwf_releases/b2o-40t1.01-Source.tar
cd b2o-40t1.01/
mkdir build
cd build/
cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/metapp/b2o/40t1.01/gnu/
-DCMAKE_MODULE_PATH=$HOME/metapp/ecbuild/2.4.0/gnu/share/ecbuild/cmake/
-DLIBEMOS_PATH=$HOME/metapp/libemos/4.4.2/gnu/
-DECKIT_PATH=$HOME/metapp/eckit/0.14.0/gnu/
-DODB_API_PATH=$HOME/metapp/odb_api/0.15.4/gnu
-DCMAKE_C_COMPILER=/soft/gfortran/bin/gcc
-DCMAKE_CXX_COMPILER=/soft/gfortran/bin/g++
-DCMAKE_Fortran_COMPILER=/soft/gfortran/bin/gfortran
make -j 4
make check
```

```
50% tests passed, 95 tests failed out of 190

Label Time Summary:
b2o    = 187.93 sec
```

```
Total Test time (real) = 188.12 sec
The following tests FAILED:
	  2 - compare-acars (Failed)
	  4 - compare-acars-mixing-ratio (Failed)
	  6 - compare-ahi-himawari (Failed)
	  8 - compare-airep (Failed)
	 10 - compare-airs (Failed)
	 12 - compare-airs-uncompressed (Failed)
	 14 - compare-amdar (Failed)
	 16 - compare-amdar-wigos (Failed)
	 18 - compare-amsr2 (Failed)
	 20 - compare-amsua (Failed)
	 22 - compare-amsub (Failed)
	 24 - compare-amv-aqua-modis (Failed)
	 26 - compare-amv-goes-13 (Failed)
	 28 - compare-amv-goes-15 (Failed)
	 30 - compare-amv-meteosat-10 (Failed)
	 32 - compare-amv-meteosat-4 (Failed)
	 34 - compare-amv-meteosat-7 (Failed)
	 36 - compare-amv-metop-b (Failed)
	 38 - compare-amv-mtsat-2 (Failed)
	 40 - compare-amv-npp (Failed)
	 42 - compare-amv-terra-modis (Failed)
	 44 - compare-atms (Failed)
	 46 - compare-ascat (Failed)
	 48 - compare-aura-omi (Failed)
	 50 - compare-aura-omi-ak (Failed)
	 52 - compare-aura-omi-macc (Failed)
	 54 - compare-aura-omi-macc-206 (Failed)
	 56 - compare-bathy (Failed)
	 58 - compare-buoy-drifting (Failed)
	 60 - compare-buoy-moored (Failed)
	 62 - compare-cris (Failed)
	 64 - compare-dribu (Failed)
	 66 - compare-drop-sonde (Failed)
	 68 - compare-envisat-meris-tcwv (Failed)
	 70 - compare-envisat-sciamachy (Failed)
	 72 - compare-ers (Failed)
	 74 - compare-gch4 (Failed)
	 76 - compare-geos (Failed)
	 78 - compare-geos-allsky (Failed)
	 80 - compare-geos-wrong-zenith-angle (Failed)
	 82 - compare-gpsro (Failed)
	 84 - compare-gmi-part1 (Failed)
	 86 - compare-gmi-part2 (Failed)
	 88 - compare-hirs (Failed)
	 90 - compare-iasi (Failed)
	 92 - compare-ims (Failed)
	 94 - compare-metar (Failed)
```

```
   96 - compare-metar-auto (Failed)
   98 - compare-metopa-gome2 (Failed)
  100 - compare-mhs (Failed)
  102 - compare-msg (Failed)
  104 - compare-mwhs (Failed)
  106 - compare-mwri-fy3c (Failed)
  108 - compare-nexrad (Failed)
  110 - compare-npp-viirs-aot (Failed)
  112 - compare-pgps (Failed)
  114 - compare-paob (Failed)
  116 - compare-pilot-land (Failed)
  118 - compare-pilot-land-bufr (Failed)
  120 - compare-profiler-american (Failed)
  122 - compare-profiler-european (Failed)
  124 - compare-qscat (Failed)
  126 - compare-rain-gauge (Failed)
  128 - compare-reo3 (Failed)
  130 - compare-resat-ak (Failed)
  132 - compare-saphir (Failed)
  134 - compare-satem-500km-merged (Failed)
  136 - compare-smos (Failed)
  138 - compare-snow (Failed)
  140 - compare-ssmi (Failed)
  142 - compare-ssmis (Failed)
  144 - compare-synop-land-auto (Failed)
  146 - compare-synop-land-bufr (Failed)
  148 - compare-synop-land-bufr-2 (Failed)
  150 - compare-synop-land-bufr-hourly (Failed)
  152 - compare-synop-land-bufr-region-6 (Failed)
  154 - compare-synop-land-bufr-with-bias-correction (Failed)
  156 - compare-synop-land-manual (Failed)
  158 - compare-synop-ship (Failed)
  160 - compare-synop-ship-abbreviated (Failed)
  162 - compare-synop-ship-bufr (Failed)
  164 - compare-synop-ship-reduced (Failed)
  166 - compare-tamdar (Failed)
  168 - compare-temp-land (Failed)
  170 - compare-temp-land-hires (Failed)
  172 - compare-temp-land-hires-huge (Failed)
  174 - compare-temp-land-hires-with-missing-significance (Failed)
  176 - compare-temp-land-zero-pressure-COPE-57 (Failed)
  178 - compare-temp-mobile (Failed)
  180 - compare-temp-ship (Failed)
  182 - compare-temp-ship-hires (Failed)
  184 - compare-terra-modis-aerosol (Failed)
  186 - compare-terra-mopitt-ak (Failed)
  188 - compare-tmi (Failed)
  190 - compare-windsat (Failed)
make install
```

COPE

```
cd $HOME/test_ecmwf_releases
tar xvf  /home/mma/mma153/work/test_ecmwf_releases/cope-40t1.01.tar
cd cope
mkdir build
cd build/
cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/metapp/cope/develop/gnu
-DCMAKE_MODULE_PATH=$HOME/metapp/ecbuild/2.4.0/gnu/share/ecbuild/cmake/
-DECKIT_PATH=$HOME/metapp/eckit/0.14.0/gnu/
-DODB_API_PATH=$HOME/metapp/odb_api/0.15.4/gnu
-DB2O_PATH=$HOME/metapp/b2o/40t1.01/gnu
-DCMAKE_PREFIX_PATH=$HOME/metapp/libemos/4.4.2/gnu/
-DCMAKE_C_COMPILER=/soft/gfortran/bin/gcc
-DCMAKE_CXX_COMPILER=/soft/gfortran/bin/g++
-DCMAKE_Fortran_COMPILER=/soft/gfortran/bin/gfortran
# make check ## BROKEN DUE TO CHANGES TO ODB SCHEMA IN THIS BRANCH
make install
```

NB:

- ODB-API tools must be included in PATH
- The ECMA.sch used by COPE is maintained in the b2o version described above.
- "mf_vertco_type" specific changes are included in the feature/mf_vertco_type branch of COPE
- scr/Cope includes the setting of the following environment variables which rely on COPE_DIR and B2O_DIR. These can be set in your Env_system file.

```
COPE_DIR=$HOME/test_ecmwf_releases/metapp/cope/develop/gnu
B2O_DIR=$HOME/test_ecmwf_releases/metapp/b2o/40t1.01/gnu/
```

```
export COPE_DEFINITIONS_PATH=${COPE_DIR}/share/cope
export ODB_SCHEMA_FILE=${B2O_DIR}/share/b2o/ECMA.sch
export ODB_CODE_MAPPINGS=${B2O_DIR}/share/b2o/odb_code_mappings.dat
export ODBCODEMAPPINGS=${B2O_DIR}/share/b2o/odb_code_mappings.dat
```