

....

ADM-Aeolus Software Testing



Jos de Kloe,
MMM : 2 maart 2009

ADM-Aeolus projecten

- L1B
 - aanleveren van atmosferische input voor en testen van L1B algorithmes en processor
- L2B
 - ontwikkelen en testen van L2B algorithmes en processor
- VAMP
 - vinden van optimale verticale sampling

ADM-Aeolus software

- E2S (simulator voor development teams)
 - producent: MDA
 - taal: matlab, c++, java, perl
- L1B (processor bij ESA grondsegment)
 - producent: MDA, nu DLR
 - taal: c++, java, fortran 77
- L2B (processor voor alle geïnteresseerden)
 - producent: ECMWF, KNMI, Meteo France
 - Fortran 90, python

ADM-Aeolus L2B software

- Eisen:
 - Draait als subroutine in ECMWF IFS
 - ==> Fortran90
 - Draait standalone op linux, unix, voor reprocessing door ESA en gebruik anderen
 - Portable, vele soorten systemen en compilers
 - ESA wil dat reprocessing met standalone versie exact hetzelfde product oplevert als subroutine versie
 - Dus identieke source code voor subroutine en standalone versies (maar identieke output is niet vanzelfsprekend!)

Soorten tests:

- FAT tests
- GSOV tests
- Unit tests
- Portability tests
- Science tests

FAT testing:

- FAT = Factory Acceptance Test
- Overtuigen van de opdrachtgever (ESA) dat alles wat ze willen hebben werkelijk geïmplementeerd is
- 100% handwerk
- Op basis van 2 documenten:
 - SRD=Software Requirements Document
 - SVVP=Software Validation and Verification Plan

GSOV testing:

- GSOV = Ground Segment Overall Validation
- Compatibility tests (compilatie, file formats, CLI/GUI interfaces)
- Integration tests:
 - automatisch verzenden van files naar ECMWF
 - triggeren van processen bij ECMWF
 - automatisch terugsturen van resultaten naar het ESA archief
- Erg formeel: veel documenten (bug rapporten door meerdere ESA teams)

Unit tests (1)

- Verifieer dat de software compileert en werkt zoals bedoeld
- Eisen:
 - Werkt automatisch
 - Vergelijk uitvoer met “verwachte” uitvoer
 - Detecteert mogelijke problemen wanneer de software op een nieuw platform of met een nieuwe compiler wordt getest
 - Stelt automatisch een rapport op voor analyse door het L2BP project team

Unit tests (2)

- Garandeert reproduceerbaarheid van de resultaten tussen verschillende machines
- Garandeert dat code wijzigingen in het ene deel van de software geen onbedoelde bijwerkingen hebben in een ander deel van de software
- Online voorbeeld:
 - http://diveintopython.org/unit_testing/

Unit tests (3): implementatie

- Voor elke module minstens 1 apart test programma (L2BP bevat ca. 100 modules)
- Aangeropen mbv een makefile
- Vergelijk uitvoer met “verwachte” uitvoer
- Makefiles en verwachte uitvoer in versie beheer (perforce)
- Uitzondering 1: **reele getallen**
 - Oplossing: eigen difftool, en speciale tags in de uitvoer vh testprogramma “REALACC(6)”
- Uitzondering 2: **afgebroken regels**

Unit tests (4): voorbeeld difftool

```
REALACC(5): z = 100.000000 200.000000 300.000000 400.000000 500.000000 ENDREALACC
REALACC(5): T_nom = 273.1000061 273.2000122 273.2999878 273.3999939 273.5000000 ENDREALACC
REALACC(5): P_nom = 1013.000000 1012.000000 1011.000000 1010.000000 1009.000000 0.000000
ENDREALACC
```

```
REALACC(5): z = 100.0000 200.0000 300.0000
400.0000 500.0000 ENDREALACC
REALACC(5): T_nom = 273.1000 273.2000 273.3000 273.4000 273.5000 ENDREALACC
REALACC(5): P_nom = 1013.000 1012.000 1011.000 1010.000 1009.000 1.e-12
ENDREALACC
```



Unit tests (5): resultaten

- De L2BP 1.40 release van afgelopen week heeft 244 unit tests
- Variërend van simpel tot compleet:
 - Aanwezigheid van data files
 - (De)allocatie en vullen van datastructuren
 - Lees en schrijf routines
 - Werking van algorithmes
 - Werking van allerhande scripts
 - Complete processor
- Controleer ook dat foutieve invoer inderdaad de **gewenste foutmelding** oplevert !



Unit tests (6): gevaren

- Ontwikkeling kan afgeremd worden omdat je bang wordt bestaande tests te breken
- Het kan een vals gevoel van veiligheid geven als de tests niet goed ontworpen zijn
 - Je test alleen dat het resultaat exact zo blijft zoals het was
 - Als er een fout in je ontwerp/algorithmie zit kom je er op deze manier nooit achter

Portability tests (1):

- Simpel: draai alle unit tests op elk platform
- L2BP software werkt op 32 en 64 bit linux voor tenminste 4 verschillende fortran compilers (pgf90, g95, gfortran, ifort)
- Ook op unix: IBM (AIX), SUN
- Problematisch op NEC
- Openstaande problemen:
 - BUFR library op 64 bits machines
 - Niet standaard Python modules

Portability tests (2):

- Waarom al die moeite?
 - Elke compiler test het systeem op zijn eigen wijze
 - Uit ervaring blijkt dat porten naar een nieuwe compiler of een nieuw platform **vrijwel altijd** nieuwe bugs of nieuwe problemen met de fortran standaard aantoont
 - Sommige compilers zijn erg tolerant, en zouden niet gebruikt moeten worden als je portable software wilt schrijven (pgf90)

Gebruikt Fortran in je voordeel

- Fortran90 modules en interface statements kunnen de **interface** van functies en routines voor je **controleren** zodat aanroep en definitie matchen !
- Fortran77 en c **kunnen dat vaak niet**
- Gebruik compiler opties om te checken op:
 - Gebruik van niet geïnitieerde variabelen
 - Array grens overschrijdingen, etc.
- Overblijvende problemen met Fortran90:
 - Pointers, (de)allocatie, optionele parameters, niet vastgelegde interfaces

Science tests (1)

- Hiervoor is het nodig om de hele keten te draaien:
 - E2S simulator
 - L1B processor
 - L2B processor
 - Diverse file conversie tools
 - Matlab analyse en plot tools
- Pass-fail criteria
- Beoordeling van de resultaten blijft altijd handwerk

Science tests (2)

- Eisen:
 - Automatisch uitvoeren
 - Reproduceerbaarheid
 - Versiebeheer (cvs, svn, perforce, clearcase, ...)
- Input:
 - Geselecteerde atmosferische data
 - Zit niet in versiebeheer
 - samengepakt in een tar file, voorzien van een versie nummer, gearchiveerd

Science tests (2)

- Software interfaces:
 - E2S: GUI voor aanmaken input files
 - E2S: GUI voor aansturen simulator
 - L1B: GUI voor aanmaken joborder file
 - L2B: commandline, input joborder file
 - Plot en analyse tools: commandline
- Uiteindelijk kunnen alle onderdelen mbv input files worden aangestuurd

Draaien vd keten van simulators

- Beschikbare tools:
 - conversie ASCII DB naar E2S input (eerst matlab, GUI, nu Python, CLI)
 - conversie ASCII DB naar L2B input (fortran, C)
 - Aanmaken van E2S scenario input files (eerst matlab, GUI, nu Python, CLI)
 - plot tool (matlab, CLI)
 - Verificatie (pass-fail), statistiek (matlab, CLI)

File

Define Input Parameters

Atmosphere: Profile Atmosphere: Ground

LIDAR Instrument Mode AOCs Errors

Manage Scenario

Current Scenario:

Create/Modify Scenario

Scenario Editor

File

Scenario Information

Scenario:

Operator Name:

Comments:

Configuration Parameters

AISP Default Parameters:

AISP Housekeeping Parameters:

AISP Error Default Parameters:

arInstrumentDetectorParameters.xml

y/lidarInstrumentLinkParameters.xml

y/lidarInstrumentMieParameters.xml

arInstrumentRayleighParameters.xml

mentCalibrationModeParameters.xml

Segment Editor

File

Segment Information

Segment ID:

Start Time [yyyy-MM-dd'T'hh:mm:ss.uuuuuu]:

Segment Parameters

AOCs Error Param

Atmosphere Profile Param

Atmosphere Cloud Param

Atmosphere Ground Param

LIDAR Instrument Mode Param

AISP Error Param

Atmosphere Profile Parameters Editor

File

Altitude [km]	HLOS Wind Vel [m/s]	Temperature [C]	Mol Ext [E-6/m]	Mol B.sca [E-6/m/sr]	Aer Ext [E-6/m/sr]	Aer B.sca [E-6/m/sr]
0	0	-1.25	73.991	8.832	0.000	0.000
0.125	0	-1.75	72.985	8.712	0.000	0.000
0.25	0	-2.15	71.955	8.589	0.000	0.000
0.375	0	-2.65	70.933	8.467	0.000	0.000
0.5	0	-3.05	69.919	8.346	0.000	0.000
0.625	0	-3.45	68.922	8.227	0.000	0.000
0.75	0	-3.95	67.951	8.111	0.000	0.000
0.875	0	-4.35	66.996	7.997	0.000	0.000
1	0	-4.75	66.066	7.886	0.000	0.000
1.125	0	-5.25	65.127	7.774	0.000	0.000
1.25	0	-5.65	64.206	7.664	0.000	0.000
1.375	0	-6.15	63.293	7.555	0.000	0.000
1.5	0	-6.55	62.379	7.446	0.000	0.000
1.625	0	-6.95	61.466	7.337	0.000	0.000
1.75	0	-7.45	60.570	7.230	0.000	0.000
1.875	0	-7.85	59.690	7.125	0.000	0.000
2	0	-8.35	58.836	7.023	0.000	0.000
2.125	0	-8.75	57.990	6.922	0.000	0.000
2.25	0	-9.25	57.152	6.822	0.000	0.000
2.375	0	-9.75	56.339	6.725	0.000	0.000
2.5	0	-10.25	55.553	6.631	0.000	0.000

Interpolate HLOS Wind Velocity Along Track Interpolate Temperature Along Track

Interpolate Mol Extinction Along Track Interpolate Mol Backscatter Along Track

Interpolate Aer Extinction Along Track Interpolate Aer Backscatter Along Track

```

3:48pm bhw034 1942 >xv
3:49pm bhw034 1943 >cp VAMP
3:49pm bhw034 1944 >o VAMP
input file: VAMP_PM1_JdeK1
Suffix = ODP
starting OpenOffice
[1] 20152
3:49pm bhw034 1945 >pwd
/usr/people/kloedej/presen
3:54pm bhw034 1946 >o Menz
input file: Menzies_Windsp
Suffix = EPS
starting ghostview
[1] 20210
3:54pm bhw034 1947 >xv
3:55pm bhw034 1948 >xv

```

L1BP HMI Main Panel

File JobOrder Edit View Log

Job Order: b/L1BP/MDA_DEL_V1.08_Runtime/etc/JobOrder.99994.xml

Processing Stage: Job Order Loaded

Generate Products

Process All

Process L0 Process L1A Process L1B/Cal

Stop

my little menu

Process.py

Run_E2S_L1B_L2B.py

VAMP

bhw034 L1BP Job Order Ed

Move Kill Xterm

EXIT Restart Dummy

Quit

cpu use

Mijn UK

Inbox f

Job Order Editor

File

Job Order

- <lplf_Job_Order>
 - <lplf_Conf>
 - <Processor_Name>: AE_RAW_L1B_WIND
 - <Version>: 01.08
 - <Order_Type>: OPER
 - <Stdout_Log_Level>: INFO
 - <Stderr_Log_Level>: NOOP
 - <Test>: false
 - <Breakpoint_Enable>: false
 - <Acquisition_Station>: Kiruna
 - <Processing_Station>: Esrin
 - <Config_Files>:
 - <Sensing_Time>
 - <Start>: 20071002_000005000000
 - <Stop>: 20071002_012018000000
 - <List_of_lplf_Procs [count="4"]>
 - <lplf_Proc>
 - <Task_Name>: IPF1_RAW_L0
 - <Task_Version>: 01.08
 - <Breakpoint>
 - <List_of_Brk_Files [count="0"]>

Browse

CreateDataInterface

Parameters

scenario Name

data directory

start Time

duration segment

N P

nbRand

mie contribution

all

AllNoise

PoissonNoise

Aocs Errors

Interpolation

Reset Counter

Running

Scenario Creation

E2S Simulator

Generate JobOrder

L1B Processors

Random Noise to L1A data

Status

File Edit Debug Desktop Window Help

Current Directory: [empty]

Shortcuts How to Add What's New

Help

Help Navigator

Workspace

Name	Value	Size	Class
ans	'/nobackup/us...	1x44	char

Current Directory Workspace

Command History

```

-- 3/03/08 12:39 PM --
-- 3/3/08 4:13 PM --
CreateDataInterface.m
pwd
cd INTERFACE
cd ..
cd INTERFACE/
1s
CreateDataInterface.m
CreateDataInterface
1s ..
1s ../CREATIONDATA/
../CREATIONDATA/DefinePaths
-- 3/3/08 4:17 PM --
pwd
cd ../INTERFACE/
./CreateDataInterface
CreateDataInterface

```

```

To
MAT

To

>> pw

ans =

/nobackup/users/matlab/MATLABT00LS_1.1/T00LS

>> cd ../INTERFACE/
>> ./CreateDataInterface
??? ./CreateDataInterface
|
Error: Unexpected MATLAB operator.

>> CreateDataInterface

SimPathL1b =

/nobackup/users/matlab/L1BP/DEL_V1.3/Installation

SimPathL1b =

/nobackup/users/matlab/L1BP/DEL_V1.3/Installation

SimPathL1b =

/nobackup/users/matlab/L1BP/DEL_V1.3/Installation

>>

```

bhw034

bhw034 bhw03 MATLAB CreateDataInterface

Probleem van GUI's:

- Enorm veel muis acties, aanwijzen, klikken, om de zeer vele velden in te vullen
- Alleen die instellingen kunnen worden gewijzigd waarvoor **een knop** is gedefinieerd in de GUI
- Het is extreem moeilijk een test **reproduceerbaar** te maken
- Aanmaken van een nieuw test scenario dat op maar enkele punten afwijkt van een vorige test, is nog steeds enorm veel werk

Test script systeem (1):

- Om al deze GUI's te omzeilen is een python script systeem ontworpen
- Eigenschappen:
 - Definieert de volgorde waarin de stappen worden gedraaid, definieert welke input data en software versies worden gebruikt
 - Simpel om opnieuw te draaien als slechts een enkele instelling verandert (bijvoorbeeld na aan/uit zetten van een ruis term)

Test script systeem (2):

- Eigenschappen (2):
 - Alle tests kunnen snel opnieuw worden gedraaid als van een of meer software pakketten een nieuwe versie beschikbaar komt
 - Alles wordt opgeslagen in versiebeheer om reproduceerbaarheid te garanderen

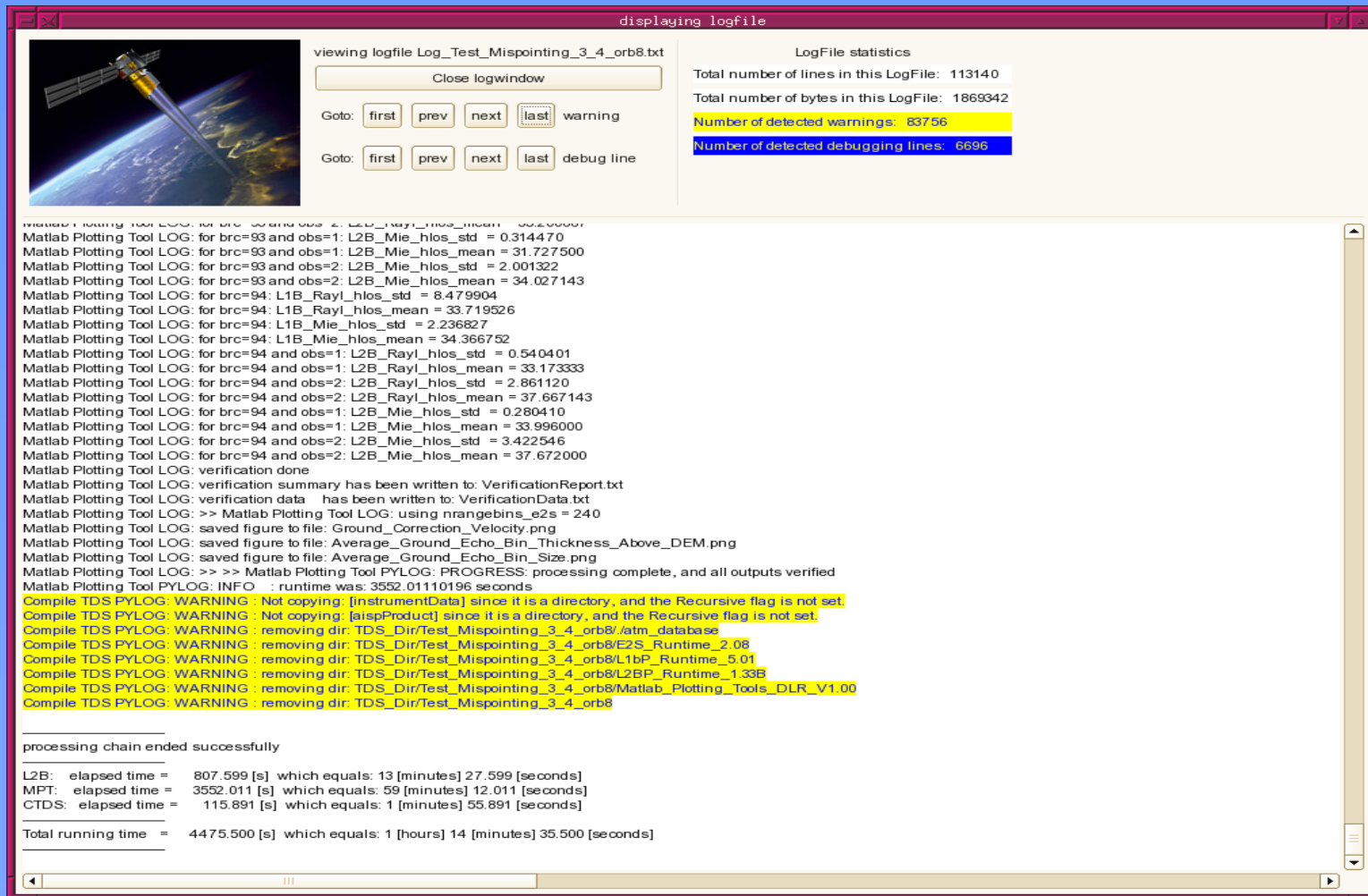
Test script systeem (3):

- Eerst is een toolbox voor basis akties opgezet:
 - Uitpakken zip/tgz files
 - Copieren van files
 - Aanmaken symbolic links en directories
 - Zetten van permissies
 - Controleren of alle nodige input files er zijn
 - Verifiëren dat alle output files aangemaakt zijn
 - Wijzigen van xml of ascii input files om de boel aan te sturen

Test script systeem (4):

- Deze toolbox wordt dan gebruikt om:
 - Elk software pakket te installeren
 - Input data en settings files te installeren
 - Bugs/errors in input file (formats) te corrigeren
 - Alle knoppen in xml files op de gewenste stand te zetten
 - Elk programma te draaien
 - Tijd en log informatie bij te houden
 - Log files te analyseren op warnings en errors

Log file view en analyse tool



displaying logfile

viewing logfile Log_Test_Mispointing_3_4_orb8.txt

Close logwindow

Goto: first prev next last warning

Goto: first prev next last debug line

LogFile statistics

Total number of lines in this LogFile: 113140

Total number of bytes in this LogFile: 1869342

Number of detected warnings: 83756

Number of detected debugging lines: 6696

```


Matlab Plotting Tool LOG: for brc=93 and obs=1: L2B_Rayl_hlos_mean = 33.23000
Matlab Plotting Tool LOG: for brc=93 and obs=1: L2B_Mie_hlos_std = 0.314470
Matlab Plotting Tool LOG: for brc=93 and obs=1: L2B_Mie_hlos_mean = 31.727500
Matlab Plotting Tool LOG: for brc=93 and obs=2: L2B_Mie_hlos_std = 2.001322
Matlab Plotting Tool LOG: for brc=93 and obs=2: L2B_Mie_hlos_mean = 34.027143
Matlab Plotting Tool LOG: for brc=94: L1B_Rayl_hlos_std = 8.479904
Matlab Plotting Tool LOG: for brc=94: L1B_Rayl_hlos_mean = 33.719526
Matlab Plotting Tool LOG: for brc=94: L1B_Mie_hlos_std = 2.236827
Matlab Plotting Tool LOG: for brc=94: L1B_Mie_hlos_mean = 34.366752
Matlab Plotting Tool LOG: for brc=94 and obs=1: L2B_Rayl_hlos_std = 0.540401
Matlab Plotting Tool LOG: for brc=94 and obs=1: L2B_Rayl_hlos_mean = 33.173333
Matlab Plotting Tool LOG: for brc=94 and obs=2: L2B_Rayl_hlos_std = 2.861120
Matlab Plotting Tool LOG: for brc=94 and obs=2: L2B_Rayl_hlos_mean = 37.667143
Matlab Plotting Tool LOG: for brc=94 and obs=1: L2B_Mie_hlos_std = 0.280410
Matlab Plotting Tool LOG: for brc=94 and obs=1: L2B_Mie_hlos_mean = 33.996000
Matlab Plotting Tool LOG: for brc=94 and obs=2: L2B_Mie_hlos_std = 3.422546
Matlab Plotting Tool LOG: for brc=94 and obs=2: L2B_Mie_hlos_mean = 37.672000
Matlab Plotting Tool LOG: verification done
Matlab Plotting Tool LOG: verification summary has been written to: VerificationReport.txt
Matlab Plotting Tool LOG: verification data has been written to: VerificationData.txt
Matlab Plotting Tool LOG: >> Matlab Plotting Tool LOG: using nrangebins_e2s = 240
Matlab Plotting Tool LOG: saved figure to file: Ground_Correction_Velocity.png
Matlab Plotting Tool LOG: saved figure to file: Average_Ground_Echo_Bin_Thickness_Above_DEM.png
Matlab Plotting Tool LOG: saved figure to file: Average_Ground_Echo_Bin_Size.png
Matlab Plotting Tool LOG: >> >> Matlab Plotting Tool PYLOG: PROGRESS: processing complete, and all outputs verified
Matlab Plotting Tool PYLOG: INFO : runtime was: 3552.01110196 seconds
Compile TDS PYLOG: WARNING: Not copying: [InstrumentData] since it is a directory, and the Recursive flag is not set.
Compile TDS PYLOG: WARNING: Not copying: [aispProduct] since it is a directory, and the Recursive flag is not set.
Compile TDS PYLOG: WARNING: removing dir: TDS_Dir/Test_Mispointing_3_4_orb8/atm_database
Compile TDS PYLOG: WARNING: removing dir: TDS_Dir/Test_Mispointing_3_4_orb8/E2S_Runtime_2_08
Compile TDS PYLOG: WARNING: removing dir: TDS_Dir/Test_Mispointing_3_4_orb8/L1bP_Runtime_5_01
Compile TDS PYLOG: WARNING: removing dir: TDS_Dir/Test_Mispointing_3_4_orb8/L2BP_Runtime_1_33B
Compile TDS PYLOG: WARNING: removing dir: TDS_Dir/Test_Mispointing_3_4_orb8/Matlab_Plotting_Tools_DLR_V1.00
Compile TDS PYLOG: WARNING: removing dir: TDS_Dir/Test_Mispointing_3_4_orb8

processing chain ended successfully

L2B: elapsed time = 807.599 [s] which equals: 13 [minutes] 27.599 [seconds]
MPT: elapsed time = 3552.011 [s] which equals: 59 [minutes] 12.011 [seconds]
CTDS: elapsed time = 115.891 [s] which equals: 1 [minutes] 55.891 [seconds]

Total running time = 4475.500 [s] which equals: 1 [hours] 14 [minutes] 35.500 [seconds]
    
```

Log file view en analyse tool



viewing logfile Log_Test_TN3_1_0051.txt

Close logwindow

Goto: first prev next last warning

Goto: first prev next last error

LogFile statistics

Total number of lines in this LogFile: 174

Total number of bytes in this LogFile: 12553

Number of detected errors: 4

Number of detected warnings: 1

```

ADB2AM LOG: Writing xml data to: /earth_explorer_header/variable_header/specific_product_header/List_of_Dsds/Dsd[14]
ADB2AM LOG: 2009-02-20T13:11:04.571000 bhw034 L2BP 01.33 [0000000862]: [P] Successfully written AMD file: AE_TEST_AUX_MET_12_20041201T000000_20041201T003000_0001
ADB2AM LOG: 2009-02-20T13:11:04.571000 bhw034 L2BP 01.33 [0000000862]: [P] Size of binary AMD file was: 40050 bytes
ADB2AM LOG: done
ADB2AM PYLOG: PROGRESS: processing complete, and all outputs verified
ADB2AM PYLOG: INFO : runtime was: 1.01107478142 seconds
L2B PYLOG: PROGRESS: Running processor: L2B
L2B PYLOG: INFO : dir exists and seems ok: L2BP_Runtime_1.33B/working/Test_TN3_1_0051
L2B PYLOG: INFO : Copied: L2BP_Runtime_1.33B/JobOrder.90001.xml to: L2BP_Runtime_1.33B/working/Test_TN3_1_0051/JobOrder.Test_TN3_1_0051.xml
L2B PYLOG ERROR : expected exactly one file but found 0 matching files for glob pattern [AE_*_ALD_U_N_1B_*].D

GUI plotting remaining STDOUT output:

GUI STDOUT: subpr pid = 861
GUI STDOUT: waiting for logfile to appear

GUI plotting remaining STDERR output:

GUI STDERR:Traceback (most recent call last):
GUI STDERR: File "/Run_E2S_L1B_L2B.py", line 2017, in ?
GUI STDERR:   if (Run_L2B): L2B.Run()
GUI STDERR: File "/nobackup/users/kloedej/versiebeheer/adm/Chain_of_Processors/Process/Process.py", line 6346, in Run
GUI STDERR:   Processor.Run(self,self.CreateJobOrder)
GUI STDERR: File "/nobackup/users/kloedej/versiebeheer/adm/Chain_of_Processors/Process/Process.py", line 2024, in Run
GUI STDERR:   if (PrepareFunction): PrepareFunction()
GUI STDERR: File "/nobackup/users/kloedej/versiebeheer/adm/Chain_of_Processors/Process/Process.py", line 6686, in CreateJobOrder
GUI STDERR:   self.L1B_dir,ScenarioDir)
GUI STDERR: File "/nobackup/users/kloedej/versiebeheer/adm/Chain_of_Processors/Process/Process.py", line 4516, in __AddJobOrderInput__
GUI STDERR: raise PrepareError
GUI STDERR: Process PrepareError

DisplayLog: added statistics to the LogFile
DisplayLog: The logfile Log_Test_TN3_1_0051.txt did contain:
DisplayLog: NumWarnings = 0
DisplayLog: NumErrors = 3
DisplayLog: NumDebugLines = 0
DisplayLog: TotalNumLines = 167
DisplayLog: NumBytes = 12292

```

Test script systeem (5):

- Test definities:
 - Zijn ook kleine python scripts van slechts enkele regels
 - Bijgehouden in versiebeheer
 - Het basis script definieert een lijst met standaard (default) waarden
 - Elke test definieert enkel de delta hierop
 - Een test kan een andere test importeren en bijvoorbeeld een enkele switch omzetten

Voorbeeld basis script (2200 regels)

```
# copy the logfiles from the executes processors into
# the main python logfile, or not?
E2S_DisplayLogFile = True
L01A1B_DisplayLogFile = True
ADB2AM_DisplayLogFile = True
L2B_DisplayLogFile = True
# send an email when done?
SendEmail = False
#SendEmail = True
# #]
# #[ Software versions
#E2S_version = "2.03"
#E2S_version = "2.04"
#E2S_version = "2.05"
#E2S_version = "2.06"
#E2S_version = "2.07" # don't use this one
E2S_version = "2.08"
#L1B_version = "1.07"
#L1B_version = "1.08"
#L1B_version = "1.09"
#L1B_version = "1.10"
#L1B_version = "1.11"
#L1B_version = "1.12"
#L1B_version = "5.00"
#L1B_version = "5.01"
L1B_version = "5.02"
```

```
# the default is to have fixed seeds for the random generator!
# The seeds are slsNoiseSeed and aocsNoiseSeed in satelliteParameters.xml
# Set UseRandomNoise to true, to switch these seeds to 0, in which case
# proper random inputs generated by using a seed from the clock will be used.
UseRandomNoise = True
#UseRandomNoise = False

AllNoiseFlag = True
PoissonNoiseFlag = True
# aocs error settings (copy defaults if true, switch off all of them if false)
# this now also sets the Roll_Error and Pitch_Error to zero
# in the file satelliteCharacterisation.xml
AocsErrorFlag = True
# if True use default values for
# mieRmsNoise/mieRmsNoiseRate/rayleighRmsNoise/rayleighRmsNoiseRate

RmsNoiseFlag = True
# if True don't simulate tripod obscuration, and don't correct for it
NoTripod = False
# if True, switch off all non-linearity effects
ForceLinear = False
# if False, set rmsLaserPulseFrequencyVariation to zero
LaserFreqVariation = True
# make all reference pulses invalid, to test the response of the software
# to invalid data (i.e. no NaN's should occur !!)
# The bug we had in dec. 2008 was a wrong default setting of this one.
Invalidate_All_Ref_Pulses = False
```

Voorbeeld test scripts

```
#!/usr/bin/env python
AtmScenarioName = "Calipso_Orbit1_Segments_1to7_and_EcmwfColloc_LowRes"
ScenarioName    = "Test_Mispointing_3_1_orb1"
PythonLogfileName = "Log_"+ScenarioName+".txt"
RangeBinSetting = "WVM2"
# import the default No Noise settings
execfile("Tests/TEST_Default_NoNoise_Settings.py")
DateTimeStart   = "2009-10-02T00:00:00.000000"
N=20
P=50
InterpolateHorizontal = True
Use_AtmosphereDataBase = True
ForceZeroAtmosphereTemperature = False
UseAtmosphereDBtiming    = True
# force an unrealistic high constant albedo everywhere
# to have valid ground echoes everywhere
ForceAlbedo = "0.8"
# disable the NWP hlos wind from the atm. db for this experiment
ForceHlosWind = "0.0" # [m/s]
# import all "run" switches
execfile("Tests/TEST_Mispointing_Run_Switches.py")
```

<-Test definitie

```
#!/usr/bin/env python
# import the 3_1_orb1 settings
execfile("Tests/TEST_Mispointing_3_1_orb1.py")
AtmosphereScenarioName = \
"Calipso_Orbit2_Segments_1to7_and_EcmwfColloc_LowRes"
ScenarioName    = "Test_Mispointing_3_1_orb2"
PythonLogfileName = "Log_"+ScenarioName+".txt"
DateTimeStart   = "2009-10-02T00:43:52.000000"
# import all "run" switches
execfile("Tests/TEST_Mispointing_Run_Switches.py")
```

Delta ->

Ondervonden problemen:

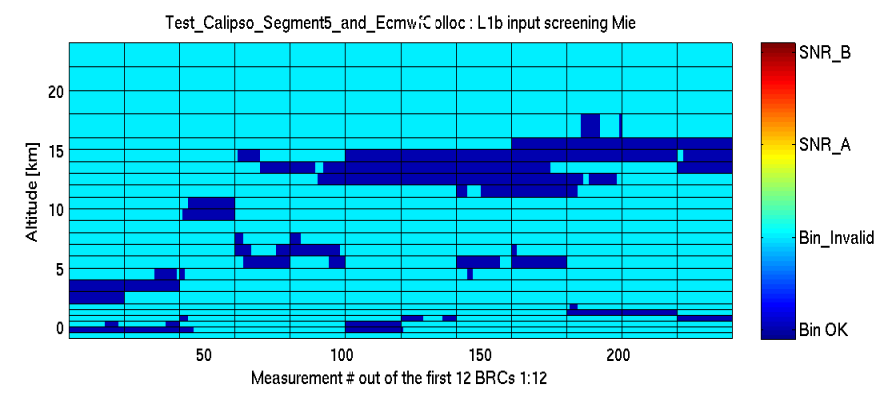
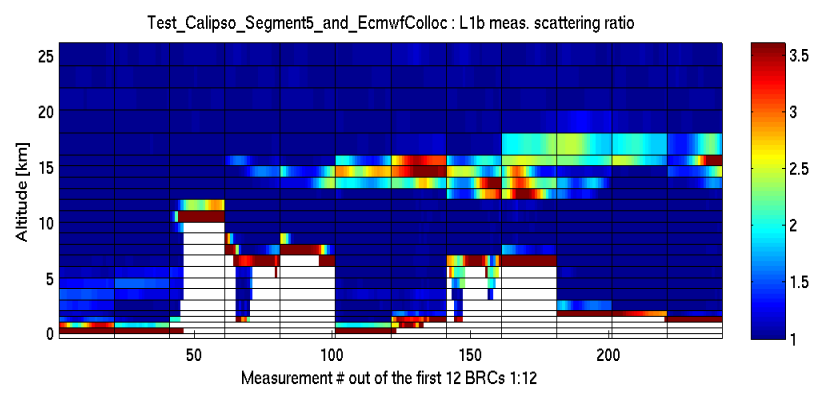
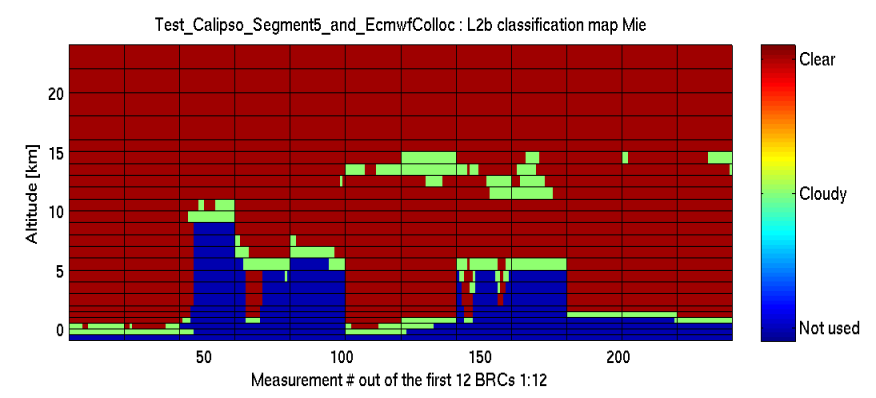
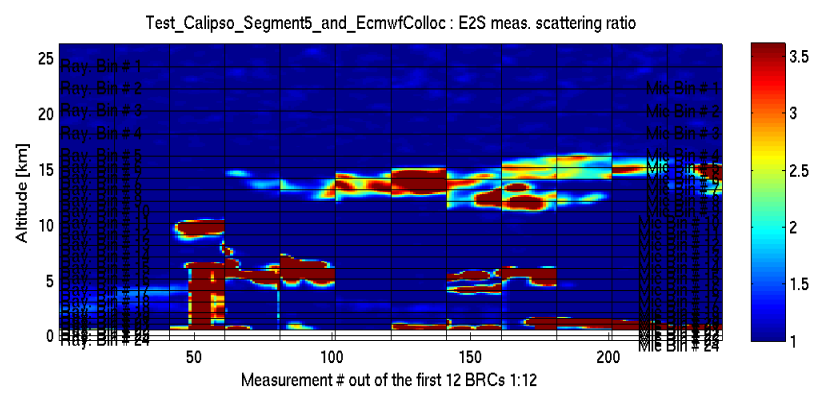
- Afhankelijk van software van derden met alle bijbehorende bugs
- Wijzigingen niet altijd goed gedocumenteerd en/of gecommuniceerd
- Geen direct contact met MDA ontwikkelaars (keuze van ESA)
- Systeem verandert te vaak. Er is nog geen gelegenheid geweest alle test resultaten goed te analyseren

••••

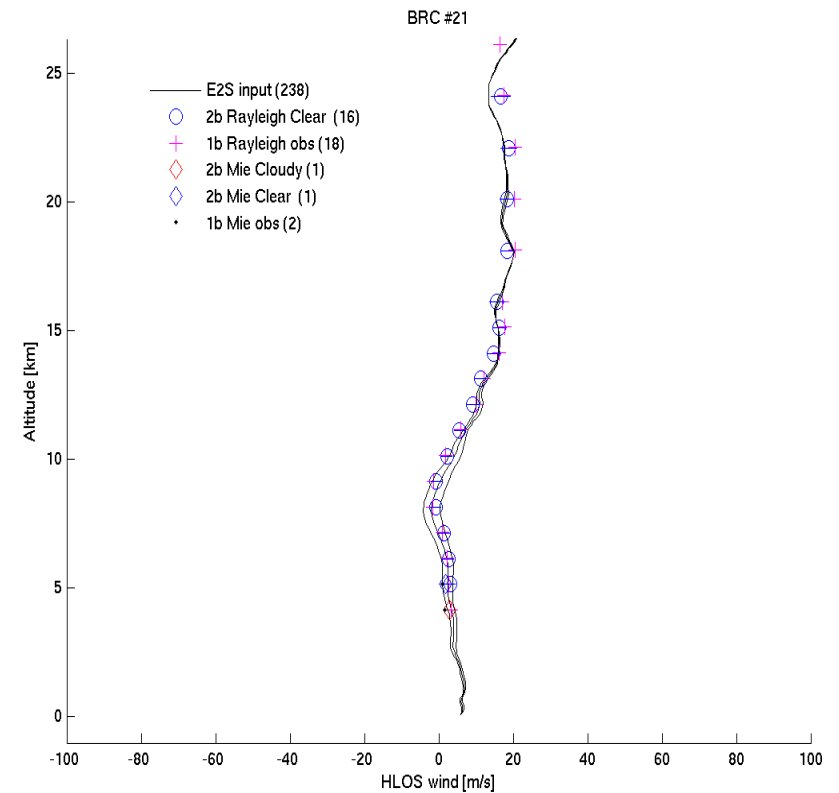
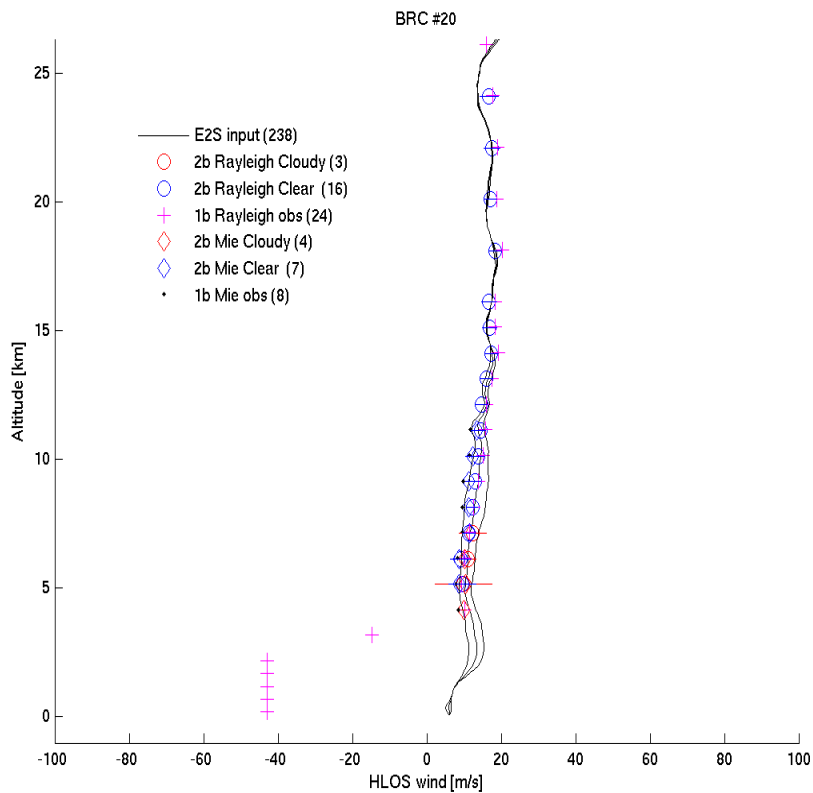
positief:

- Het hele script systeem werkt feilloos
- Er zijn 116 tests gedefinieerd die het hele systeem behoorlijk op de pijnbank leggen
- Dankzij het log systeem is vaak meteen duidelijk waar het probleem zit als er iets niet werkt
- In individuele componenten zitten nog bugs, die we met dit systeem kunnen aantonen

Een mooi plaatje tot besluit (1)



Een mooi plaatje tot besluit (2)





Einde

- vragen ?

