OMPS Nadir App Testbed

V2.0 under development

Jason Li & Colin Seftor

09/24/2015 OMPS STM

APP Sandbox

(unit test)

SIPS

(full-scale production)

What if you want to run months worth of data & fine-tune an APP as you go along?

What is the nadir testbed?

It is a customized system/framework for OMPS nadir that let users experiment their ideas by running L1/L2 apps over sufficient quantity of data, on orbit-by-orbit basis. It runs on TLCF and has no entanglement with the SIPS production system.

Keywords:

- *Experiment*: make changes in the apps in any which way, then rerun.
- *Sufficient Data*: weeks to months worth of data.
- *Disconnected* from SIPS: no production rules involved.
- *Customized* system: not a turn-key solution and not a generic framework!

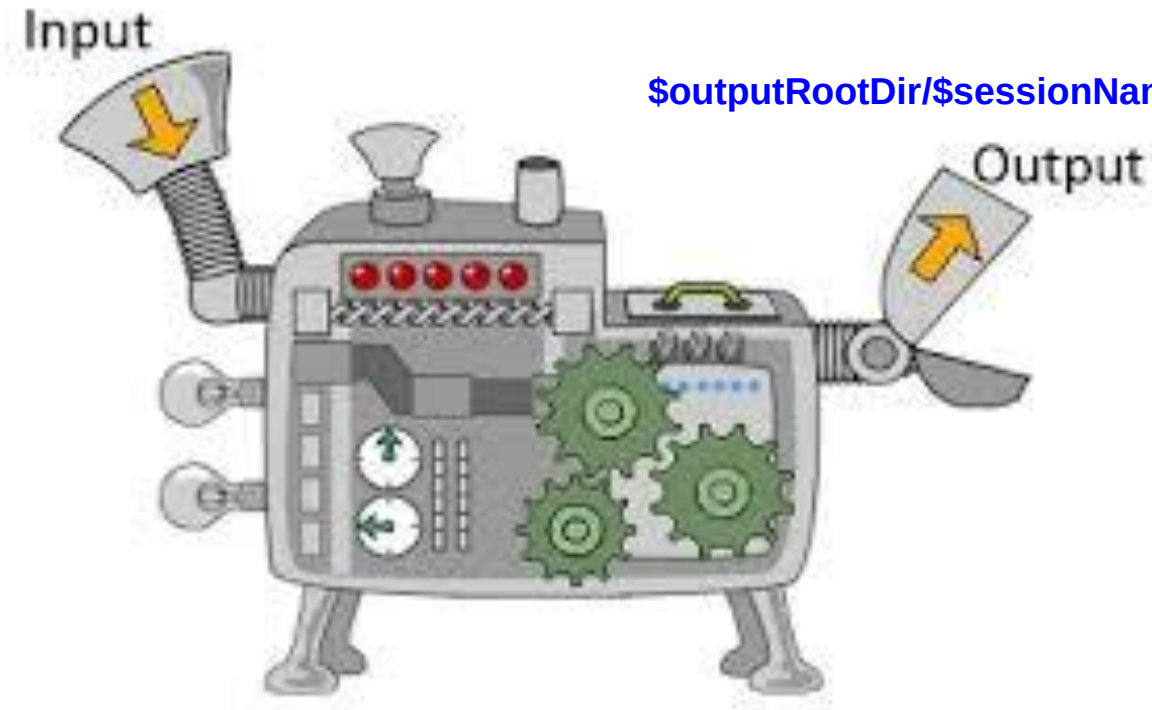
How it works - overview

#1: Organize flow of data: input, within the chain, output

#2: Construct control/PCF files

#3: Manage “oddities”

$inputRootDir/ESDT_NAME/YYYY/MM/DD



$outputRootDir/$sessionName/ESDT_NAME/YYYY/MM/DD

single APP *or* multiple of APPs in a Chain

How it works - omps_chain directory layout

```
omps_chain
    ---bin                          → where python scripts live

    ---dynamic_tables               → tables to be used
       ---NMEV-L1B

    ---templates                    → PCF/control file templates
       ---pcf_file
       ---control_file

    ---app                          → apps sit in their own containers
       ---NMEV-L1B
          - NMEV-L1B-1.5.2
          - Nadir-L1B-EV-1.5.3
          - L1B-Shr-1.0.7
         ...

       ---NMTO3-L2
          - NMTO3-L2-2.0.4
          - JLi_Fortran_Toolkit-1.0.0
          - JLi_Python_Toolkit-0.0.2
         ...

    ---incoming                     → gateway to the outside world
```

How it works - bin

cd bin

```
run_testbed_chain.py sessionConig.yml
```

sessionConfig.yml

```
SessionName: "v2_softcal"
InputRootDir: "/omi/live/dd/85605"
StartDate: "2013-04-01"
EndDate: "2013-04-30"
OutputRootDir: "/scratchdev/jyli/testbed_sessions"
```

Division of labor between your driver program and omps_base.chain.py:

run_testbed_chain.py: is a driver program, mine in this case!
(1) bring inputs from outside into the chain.
(2) calls the omps_base_chain.py.
(3) store outputs to $OutputRootDir

omps_base_chain.py: core algorithm
(1) construct appropriate control or PCF files
(2) run apps
(3) manage the data flow within the chain (set union operations)

How it works - incoming

```
omps_chain
    ---bin

    ---dynamic_tables
       ---NMEV-L1B

    ---templates
       ---pcf_file
       ---control_file

    ---app
       ---NMEV-L1B
          - NMEV-L1B-1.5.2
          - Nadir-L1B-EV-1.5.3
          - L1B-Shr-1.0.7
         ...

       ---NMTO3-L2
          - NMTO3-L2-2.0.4
          - JLi_Fortran_Toolkit-1.0.0
          - JLi_Python_Toolkit-0.0.2
         ...

    ---incoming  ------------------------>  gateway to the outside world
```

How it works - incoming

Sample chain control file ompsChain_control.txt constructed by the driver program

```
Input Files:
  LEAPSECT:
  - leapsec.dat.2013110816728
  NP_DRK:
  - OMPS-NPP-NP_DRK-p058-v1.0-2013m0901t192837-o09569-2013m0905t171816.h5
  NP_L1A_EV:
  - OMPS-NPP-NPP-NP_L1A_EV-v1.0-2013m0901t174708-o09568-2013m0905t153003.h5
  TC_DRK:
  - OMPS-NPP-TC_DRK-p058-v1.0-2013m0901t192837-o09569-2013m0905t172309.h5
  TC_L1A_EV:
  - OMPS-NPP-TC_L1A_EV-v1.0-2013m0901t174708-o09568-2013m0905t152956.h5
  UTCPOLET:
  - utcpole.dat.2013110816728

Runtime Parameters:
  EndTime: 2013-09-01T19:28:37.000000Z
  OrbitNumber: 09568
  ProcessingCenter: OMPS Nadir Testbed
  StartTime: 2013-09-01T17:47:08.000000Z
```

How it works - template

```
omps_chain
    ---bin

    ---dynamic_tables
       ---NMEV-L1B

    ---templates
            ----pcf_file
            ----control_file

    ---app
       ---NMEV-L1B
          - NMEV-L1B-1.5.2
          - Nadir-L1B-EV-1.5.3
          - L1B-Shr-1.0.7
         ...

       ---NMTO3-L2
          - NMTO3-L2-2.0.4
          - JLi_Fortran_Toolkit-1.0.0
          - JLi_Python_Toolkit-0.0.2
         ...

    ---incoming
```

How it works - template

A sample control file: control_file/NMTO3-L2_ctl.txt

```
---
Input Files:
  400410:
    - NVAL_LUT_OMPS.h5
  400411:
    - DNDX_LUT_OMPS.h5
  LEAPSECT:
    - leapsec.dat.2015041758433
  NM-L1-ANC:
    - OMPS-NPP_NM-L1-ANC_v1.8_<DataDate>_<Orbit>_0000m0000t000000.h5
  NMEV-L1B:
    - OMPS-NPP_NMEV-L1B-p000_v1.8_<DataDate>_<Orbit>_0000m0000t000000.h5
  TPV-ALT:
    - OMPS-NPP_TPV-ALT-v000-005_2014m0721t133000_s6482001efa8c61dcc29b8b4636d755bf.yml
  TPV-NMTO3-L2:
    - OMPS-NPP_TPV-NMTO3-L2_2012m0401t010229_o02209_2015m0422t141200.yml
  UNDEF:
    - LUT/NVAL_LUT_OMPS.h5
    - LUT/DNDX_LUT_OMPS.h5
  UTCPOLET:
    - utcpole.dat.2015041758433
Output Files:
  NMTO3-L2:
    - OMPS-NPP_NMTO3-L2_v2.1_<DataDate>_<Orbit>_<ProductionTime>.h5
  NMTO3-L2-Ozone-Image:
    - OMPS-NPP_NMTO3-L2-Ozone-Image_v2.1_<DataDate>_<Orbit>_<ProductionTime>.png
Runtime Parameters:
  APPVersion: 2.0.3
  CLOUDPRESSURESOURCE: '"OMCLDRR"'
  EndTime: <EndTime>
  Instrument Name: NPP_OMPS-TC
  InstrumentName: '"OMPS"'
  LOCALVERSIONID: '"RFC1321 MD5 = not yet calculated"'
  OPERATIONMODE: '"Test"'
  OrbitNumber: <Orbit>
  PGE: NMTO3-L2
  PGEVERSION: '"2.0.3"'
  PGEVersion: 2.0.3
  ProcessLevel: '"2"'
  ProcessingCenter: <ProcessingCenter>
  ProcessingHost: <ProcessingHost>
```

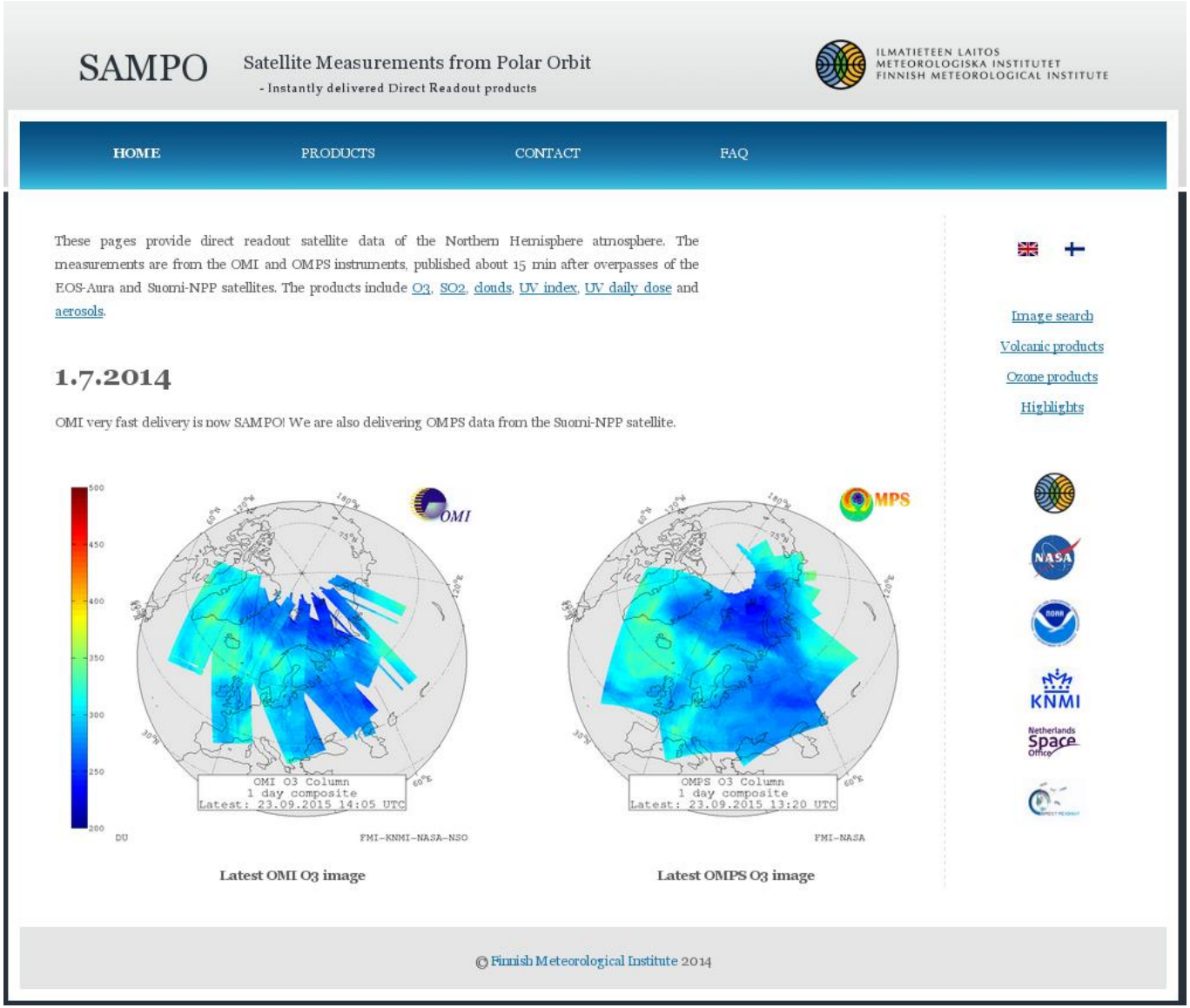
How it works - dynamic tables

```
omps_chain
    ---bin

    ---dynamic_tables  ---------------> tables to be used
            ---NMEV-L1B
            ...

    ---templates
       ---pcf_file
       ---control_file

    ---app
       ---NMEV-L1B
          - NMEV-L1B-1.5.2
          - Nadir-L1B-EV-1.5.3
          - L1B-Shr-1.0.7
         ...

       ---NMTO3-L2
          - NMTO3-L2-2.0.4
          - JLi_Fortran_Toolkit-1.0.0
          - JLi_Python_Toolkit-0.0.2
         ...

    ---incoming
```

How it works - dynamic tables

TC_SDR_EV_CHAIN/inventory.yml

```
APP Name: TC_SDR_EV_CHAIN

Input Files:
   NPP-OPF:
      - OMPS-NPP-NPP-NPP-OPF-v026-049-2012m0611t215401-sc9584cc5e542de766679ae9c5ed166a0.h5
   TC_BPS:
      - OMPS-NPP-NPP-TC_BPS-v012-068-2011m0906t181030-s73325289f8b19ebd218b87cc9fec5eeb.h5
   TC_STB:
      - OMPS-NPP-NPP-TC_STB-id050-v001-001-2011m0628t151843-sb9e8b0d617c3bfd74cfb7b7b2af6b0bd.h5
   TC_CBC:
      - OMPS-NPP-TC_CBC-v023-017-2012m0330t221534-sfea594cd9828e0606f352b06668380f0.h5
   TC_IRF:
      - OMPS-NPP-TC_IRF-v022-088-2012m0323t192223-s853e923d61389c691d35279072de8b7a.h5
#     - TC_IRF_20130304.h5
#     - TC_IRRAD_DichroicTesting_V1.h5
#     - TC_IRRAD_DichroicTesting_V2.h5
#     - TC_NEW_IRF_cbc_shift.h5
#     - TC_NEW_IRF_015nm_shift.h5
#     - TC_NEW_IRF_025nm_shift.h5
   TC_RAD:
      - OMPS-NPP-TC_RAD-v006-052-2011m0215t224554-s160fe7664302850fa7f398b2d1848efd.h5
#     - TC_RAD_DichroicTesting_V3.h5
#     - OMPS-NPP-TC_NEW_RAD_025.h5
#     - OMPS-NPP-TC_NEW_RAD_035.h5
#     - TC_NEW_RAD_COEF_cbc_shift.h5
#     - TC_NEW_RAD_COEF_015nm_shift.h5
   TC_SLT_EV:
      - OMPS-NPP-TC_SLT_EV-v031-005-2013m0114t202942-s5eea9f2a9094249f23eb45427abb238f.h5
   TC_SRG:
      - OMPS-NPP-TC_SRG-v006-052-2011m0215t224554-safa7642ef612b81a4d39b9b8c30261a9.h5
   TP:
      - OMPS-NPP-TP-v023-035-2012m0409t194716-s7b4a9af5fee69085f3bcb35876a54040.h5
```

Spinoffs - Direct Readout Application

http://omivfd.fmi.fi/ij2j3f0/index.html
Finnish Meteorological Institute (FMI)
Volcanic Products from Satellite Measurements from Polar Orbit (SAMPO)