Various Tips and Tricks for Efficient Use of ESARAD on Unix and Windows

Hans Peter de Koning
Hans-Peter.de.Koning@esa.int
ESA/ESTEC D/TOS-MCV

Contents

- Some issues with large models
- Thermal analysis compared with software engineering
- Configuration control
Why do we have large models?

- Some spacecraft/systems (Envisat, Metop, ISS) are very large themselves, therefore demand large thermal models.
- Aim to reduce costly hardware testing through increased software simulation.
- Every next generation computing platform (1.5 years) can handle larger problems.

Problems of large models

- Objective is to perform adequate thermal analysis ...
  But in practice much time is spent on ‘data processing’.
- Ensuring consistency between partners (prime, sub, customer).
- Managing analysis cases, runs, results, model versions.
- Interpreting / post-processing huge amounts of results.
- Long analysis cycle time due to considerable CPU/elapsed time.

Very similar to software development problems
Therefore use Software Engineering methods and technology.
**Make use of Software Engineering best practices**

- Define data once, reference thereafter
  - No duplication of definitions
  - E.g. ESARAD INCLUDE, INCLUDE_MODEL
- Hierarchical breakdown & encapsulation to manage complexity
- Scripts for automated generation where possible
  - Caveat: How to keep scripts portable across platforms?
- Establish/implement working procedures
  - Include naming conventions
  - Standardise directory structures
- Could use configuration control tool CVS (www.cvshome.org)
  - But significant effort to set-up

**make & makefiles**

- Standardise on GNUmake to ensure portability
- Some Unix/Windows incompatibilities:
  - echo works differently under Unix shell and Windows .bat, e.g.
    - Unix: echo "DELETE_MODEL \"mymodel\";" | esrda
    - Windows: echo DELETE_MODEL "mymodel"; | esrda
  - Unix symbolic links and Windows shortcuts work differently
    - Recommendation: avoid symbolic links / shortcuts
- As part of migration guidelines we are currently detailing how-to move make from Unix to PC (and vice-versa)
  - Will provide a worked example which works on both Unix and Windows
Portability Unix - Windows (1)

- shell scripts & make
  - Can convert simple Unix shell scripts to .bat / .cmd files
    - OK for one-time conversion, cumbersome to maintain in sync
  - GNUmake available on Windows
  - Cygwin: complete Unix shell/utilities (including make) on top of Windows, see http://sources.redhat.com/cygwin
  - Demo follows
- On Windows NT inherent limit of 2GB addressable memory per process, versus 4GB for 32-bit Unices
  - May have an impact on running ESATAN file formatting for large number of orbit positions / moving geometry (time-varying GRs)

Portability Unix - Windows (2)

- ESARAD on PC supports forward and backward slash in directory paths
  - e.g. INCLUDE "properties/bol.erg"
  - Use Unix style directory paths for portability
  - Use relative directory paths only - i.e. avoid Windows drive letters
  - Define absolute directory paths in ESARAD preferences .esaradrC
Use of Preferences (.esaradc) in combination with HOME

file $HOME/.esaradc (Unix) or %HOME%\esaradc (Windows):
esarad_user_model_directory = d:\users\hanspeter\EsaradModels
esarad_user_files_local = true
esarad_auto_save = 0
esarad_temp_files = false
gem_temp_files = false
vis_complex_cutting = false
esarad_search_path = NULL
esarad_site_model_directory = NULL
esarad_sys_fonts = true
esarad_font_size = 0

Switch project, model, campaign using HOME and .esaradc

• Set HOME environment variable for project/model/campaign
  export HOME=/home/user1/project1/esarad

  • .esaradc in $HOME
    esarad_user_model_directory = /home/user1/project1/esarad
    esarad_user_files_local = false
    esarad_auto_save = 0
    esarad_temp_files = false
gem_temp_files = false
    vis_complex_cutting = false
    esarad_search_path = /home/user1/project1/esarad/include
    esarad_site_model_directory = /home/esarad/esarad_database
    esarad_sys_fonts = true
    esarad_font_size = 0
Structured assembly of geometry and kernel files (1)

- Skeleton mymodel_myconfiguration_mypropenv.erg:

```plaintext
BEGIN_MODEL mymodel_myconfiguration_mypropenv
/* get property environment, e.g. BOL or EOL */
INCLUDE "mypropenv.erg" /* from myproject/esarad/include */
/* get configuration parameters, e.g. "SA_stowed", "SA_deployed" */
INCLUDE "myconfiguration_parameters"
/* get geometry definition */
INCLUDE "mymodel_geometry"
END_MODEL
```

Structured assembly of geometry and kernel files (2)

- Skeleton mymodel_myconfiguration_mypropenv_init.erk:

```plaintext
BEGIN_MODEL mymodel_myconfiguration_mypropenv
/* define all user procedures, including compute_orbit */
INCLUDE "mymodel_procedures.erk"
/* declare and define default setting for all variables */
INCLUDE "mymodel_init.erk"
END_MODEL
```

- Skeleton mymodel_myconfiguration_mypropenv_myfluxcase.erk:

```plaintext
BEGIN_MODEL mymodel_myconfiguration_mypropenv
INCLUDE "mymodel_procedures.erk"
/* perform myfluxcase specific calculations */
...
/* run ESARAD calculate_x procedures */
compute_orbit( param1, param2, ..., startpos, endpos, ... );
END_MODEL
```
Advantages of structured assembly approach

- Can split orbit calculation into a sequence of partial orbits
  - Very useful for machines with limited resources
  - Potential reduction of elapsed time on multi-processor machines
- Can use the same set-up for along-orbit or per-orbit-position visualisation with correct pointing/articulation
  - Therefore consistency between kernel and visualised geometry
- No more maintenance of **DELETE variable** statements needed

Syntax colouring text editor

- On PC Windows found a very good shareware editor: TextPad
  - www.textpad.com (shareware: USD27 for one license - USD120 for 5)
  - works fine (and fast) with very large files (>1 million lines)
- On Unix (x)emacs and nedit
  - www.gnu.org and www.nedit.org (freeware)
- Made syntax definition for ESARAD and ESATAN
  - demonstration follows
    - ready for TextPad - in the works for nedit
  - Volunteers for emacs syntax files?
(1) Concerning computer platform usage for thermal analysis
What trends do you expect in the coming 3~5 years?
   – (a) Unix workstations & Unix computation/file servers
   – (b) PC workstations & Unix computation/file servers
   – (c) PC workstations & PC servers

(2) If PCs: Win98 / WinNT4 / Win2000 ?
(3) Is your organisation using Linux workstations/servers ?
(4) Does your organisation have a policy on Linux usage ?