Appendix L

Improve thermal analysis process with Systema V4 and Python

Alexandre Darrau
(Airbus Defence and Space, France)
Abstract

When performing analyses, thermal engineers follow a methodology to ensure results quality and traceability. However, some checking or/and post-processing operations are still manually done or are performed later in the analysis process, leading to error and time wasting. The purpose of this presentation is to introduce how the Airbus Defence & Space Thermal Engineering department in Toulouse is working to overcome these difficulties using new Systema V4 functions and Python technology. An example for each thermal analysis stage is going to be presented to illustrate.
**Improve thermal analyses process with SystemaV4 + Python**

Alexandre Darrau, Jean-Baptiste Bernaudin
Thermal Engineering - Airbus Defence & Space – Thermal Team

---

**PLAN**

- Objectives & Strategy
- Application on Thermal analysis process

01 - Model Geometry
02 - Model Pre-conditionning
03 - Model Conditionning
04 - Runs
05 - Model PostProcessing

**Systema**
- Python API
- PySDS
- PostProcessing Kernel

---

WARG
GIMLI
RADYS
RADAGAST
TAPAS
POWERANGER
Objectives & Strategy (1/2)

Objectives: Support thermal engineer!

- **Prevent** time wasting + manual operations ➔ Automatisation (helper tools…)
- **Ensure** thermal analysis quality ➔ Check at each analysis stage
- **Remove** industrials softwares borders ➔ Object-Oriented + Modular approach
- **Standardize** data presentations.

Automatic tools will **never** replace engineer thermal but **only optimize** its time!

Objectives & Strategy (2/2)

**Strategy:**

- Set global methodology to pinpoint thermal engineer needs
- For each need, define a method to apply.
- When a tool is needed:
  - Use object oriented approach
  - Split data treatment from format
  - Category: modelling helper tool, checker, analysis tool

Thermal Team Tool

**SYSTEMA - API**

Model/Meshing../PostPro SDS

**Thermal Team Python Modules**

⇒ Export Data
PLAN

• Objectives & Strategy
• Application on Thermal analysis process

01 - Model Geometry
02 - Model Pre-conditioning
03 - Model Conditionning
04 - Runs
05 - Model PostProcessing

WARG GIMLI
RADYS
RADAGAST
TAPAS
POWERANGER

Systema
Python API
PySDS
PostProcessing Kernel

Model Geometry

GIMLI: Generation MLI

- Modelling Helper Tool
- Automatize MLI GMM and TMM model generation

CAO

SIMPLIFY MODEL

GMM
TMM (.d)
Mesh coatings
Nodes Boundaries GL GR

11/03/2015
Model Geometry

**WARG: Waveguide Rapid Generator**

- Modelling Helper Tool:
  - Automatize waveguide GMM and TMM model generation

CAO Neutral Line + Support Pos

- GMM
- TMM (.d)
- Mesh
coatings

**PLAN**

- Objectives & Strategy
- Application on Thermal analysis process

01 - Model Geometry

02 - Model Pre-conditionning

03 - Model Conditionning

04 - Runs

05 - Model PostProcessing

**Systema**

- Python API
- PySDS
- PostProcessing Kernel

This document and its content is the property of Airbus Defence and Space. It shall not be communicated to any third party without the owner's written consent.
Model Preconditionning

**RADYS: Radiative synthesis**

- **Checker tool**: Check radiative cases data and generate report before run!

### Systema API + Thermal Team Module

#### RADYS

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model name</td>
<td>RADYS</td>
</tr>
<tr>
<td>Run parameters</td>
<td></td>
</tr>
<tr>
<td>Run parameter</td>
<td></td>
</tr>
<tr>
<td>Run parameter value</td>
<td></td>
</tr>
<tr>
<td>Run parameter units</td>
<td></td>
</tr>
<tr>
<td>Plant mass parameters</td>
<td></td>
</tr>
<tr>
<td>Plant mass parameter</td>
<td></td>
</tr>
<tr>
<td>Plant mass parameter</td>
<td></td>
</tr>
<tr>
<td>Plant mass parameter</td>
<td></td>
</tr>
<tr>
<td>Plant mass parameter</td>
<td></td>
</tr>
<tr>
<td>Radiation parameters</td>
<td></td>
</tr>
<tr>
<td>Radiation parameter</td>
<td></td>
</tr>
<tr>
<td>Radiation parameter</td>
<td></td>
</tr>
<tr>
<td>Radiation parameter</td>
<td></td>
</tr>
<tr>
<td>Radiation parameter</td>
<td></td>
</tr>
<tr>
<td>Solar flux parameters</td>
<td></td>
</tr>
<tr>
<td>Solar flux parameter</td>
<td></td>
</tr>
<tr>
<td>Solar flux parameter</td>
<td></td>
</tr>
<tr>
<td>Solar flux parameter</td>
<td></td>
</tr>
<tr>
<td>Solar flux parameter</td>
<td></td>
</tr>
<tr>
<td>Mass parameters</td>
<td></td>
</tr>
<tr>
<td>Mass parameter</td>
<td></td>
</tr>
<tr>
<td>Mass parameter</td>
<td></td>
</tr>
<tr>
<td>Mass parameter</td>
<td></td>
</tr>
<tr>
<td>Mass parameter</td>
<td></td>
</tr>
</tbody>
</table>

**11/03/2015**

**AIRBUS DEFENCE & SPACE**
Model Conditionning

RADAGAST: RADiAtive GebbArt Support Tool
- Checker tool: check gebbart factor consistency on geometry.

+ Gebbart Data

11/03/2015
Model Conditionning

RADAGAST: RADiAtive GebbArt Support Tool

- Check gebbart factor consistency on geometry.

Face active see inactive one!
Model PostProcessing

POWERANGER: POWEr Range GEneratoR
- Tool for thermal analysis
  - Perform power budget on groups after thermisol runs.
  - Steady-State and Transient cases on selected times.
  - Taking into account GL,GR variations and Edges elements.
  - Having graphical views of power exchanges and table synthesis.

Groups file
Solver File H5
SYSTEMA POSTPROCESSING + Thermal Team Module
HTML
PNG

Model PostProcessing

POWERANGER: POWEr Range GEneratoR
- Tool for thermal analysis
Model PostProcessing

**POWERANGER: POWER Range GEnerator**

- Compute $T_{min/max/ave}$/Gradients on groups **AFTER** runs
- Compute $Q_I/Q_R/Q_S/Q_A/Q_E$ on groups **AFTER** runs
- Compute Mass Balance **AFTER** runs + Manage Equipment status
- Generate automatically charts (2d and 3d horizontal bars), 3dviews
- Generate automatically Excel + HTML thermal reports

---

**TAPAS: Thermal Analysis PostProcessing Airbus Satellite**

- Compute $T_{min/max/ave}$/Gradients on groups **AFTER** runs
- Compute $Q_I/Q_R/Q_S/Q_A/Q_E$ on groups **AFTER** runs
- Compute Mass Balance **AFTER** runs + Manage Equipment status
- Generate automatically charts (2d and 3d horizontal bars), 3dviews
- Generate automatically Excel + HTML thermal reports

---

**SYSTEMA POSTPROCESSING + Thermal Team Module**

- HTML
- PNG
- XLS
Model PostProcessing

TAPAS:

![TA-PAS screenshot](image)

**ENTITIES INFORMATION**

<table>
<thead>
<tr>
<th>Entities</th>
<th>Temperature</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI_Buson</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI_Cables</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Share</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI_Aisland</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI_Pad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI_Inputs_T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI_Inputs_T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI_HeatInput</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI_Radiator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI_Total_W</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI_Heating</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI_Airot</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

11/03/2015
Model PostProcessing

TAPAS:

Improve thermal analysis process with Systema V4 and Python

11/03/2015

Model PostProcessing

TAPAS:

3D VIEWS DEFINITION

11/03/2015
Improve thermal analysis process with Systema V4 and Python
Improve thermal analysis process with Systema V4 and Python

TA-PAS

Temperature.png

TA-PAS

Temperature.png
CONCLUSIONS

- Thanks to SYSTEMA API, it is possible to:
  - Optimize time model/meshing creations
  - Easily check thermal model.
  - Plotting user data on mesh.

- Thanks to Python, it is possible to:
  - Create simple tools without deep software engineering knowledge.
  - Wrap Systema API
  - Design tools with oriented object approach.
  - Create user friendly tools thanks to existing packages.

- Collaboration with Systema Team to improve existing functions and create new ones:
  - Materials data to be integrated in API
  - Implementing new box in postprocessing library
  - ....
QUESTIONS ?