Robust industrial model exchange between ESARAD and THERMICA with STEP-TAS

Hans Peter de Koning, Simon Appel, David Alsina Orra
(ESA/ESTEC D/TOS-MCV)

with some contributions from Simulog and Graitec

Topics

• Brief recap of STEP-NRF and STEP-TAS protocols
• Why TASverter?
• Scope and purpose of TASverter
• Development approach
• Updates to NRF and TAS protocols
• Current status
• Outlook
Recap STEP-NRF (1)
(Network-model Results Format)

- Generic, discipline-independent exchange of models & results
  - Model definition, using a discrete network representation
    • Supports model/submodel hierarchy
  - Results data, produced in analysis, test or operation
  - Meta-data, which records details of actual analysis, test or operation performed
- Only discrete observations
  - I.e. sampled results at discrete locations for discrete states of the model / object
    under observation, no support for continuous fields or similar
- Any property value has explicit (physical) quantity and unit
- Data model designed to cope efficiently with large amounts of results
  - Built-in support for scalar, vector, matrix, tensor data

Recap STEP-NRF (2)
(Network-model Results Format)

Central NRF data structure is the ‘datacube’
- each element of the cube is a scalar, vector or tensor property for a specific (item, quantity, state)
- state quantity is normally time or frequency
- ’simple’ and ’advanced’ SUBTYPES, respectively for literal and generalised functional property values
Recap STEP-TAS
(Thermal Analysis for Space)

• STEP-based application protocol
  – Initial scope: Exchange of thermal-radiative models for space, including rigid body kinematics and orbit / attitude / orientation specification
    • Geometry represented by bounded face model with minimal topology (compatible with AP 203 CC4)
  – Extended scope: Exchange of thermal lumped parameter network models
    • Targeting exchange between various SINDAs, ESATAN
  – STEP-TAS is a pure superset of STEP-NRF (Used as ‘integrated resource’)
  – Developed since 1996 mainly on ESA funding, supported by CNES, NASA
    • Originally full ARM / AIM according to ISO TC184/SC4 procedures
    • Accompanying STEP-TAS library (C and F77 API) provided since 1998
    • Many tools implemented prototype or industrial beta converters

Why TASverter?

• STEP-TAS converters did not deliver industrial solution up to now
  – Only ESARAD, THERMICA and Thermal Desktop have STEP-TAS exchange included in industrial releases
  – Exchange is slow, often not reliable, and fails for large models
• Existing STEP-TAS architecture had too many layers
  – Bad performance (CPU / elapsed time)
  – Inefficient memory usage, huge memory requirements
  – Expensive to verify and maintain
  – Difficult to distribute on multiple platform/compiler combinations
• However principle of providing protocol + library was very good and should be retained!

'STEP-TAS high level API'
(C and F77)
STEP-TAS ARM
STEP-TAS AIM
SDAI-C
(COTS, late binding)
Vendor specific repository handler (COTS)
**Scope and purpose of TASverter**

- TASverter is an initiative of ESA/ESTEC D/TOS-MCV to:
  - Offer users finally a properly working solution for exchange of thermal models
    - First between major European analysis tools ESARAD and THERMICA
  - Remove complicated dependency on (at least) 4 developers
    - STEP-TAS and STEP library developers, Tool X and Tool Y developers
  - Produce a fully functional (open source) framework for validation and verification of STEP-based data exchange protocols and implementations
  - Lay a solid basis for the future
    - Low threshold for implementation
    - Maintainable and cost-effective
    - Ensure long term availability (no dependence on closed 3rd party software)

**Development approach (1)**

- Implementation in pure Python (v2.2)
  - Following positive experience with earlier ‘ad-hoc’ converter developments
- Internal data storage uses STEP-TAS (ARM) data model
  - Implemented in ‘STEP-TAS Repository Handler’, which is largely generated automatically with pyExpress from the STEP-TAS EXPRESS schema
- For each supported tool/format a ‘reader’ and a ‘writer’ is created
- Full testsuite built up alongside development
  - Unit, integration and large model testcases under configuration control
- Integrated validation and fine-tuning of STEP-NRF and STEP-TAS
  - Goal is recreation of models which are understandable and editable by humans
  - Efficient update cycle is possible with pyExpress STEP-TAS library generator
**Development approach (2)**

- **Phase 1**
  - THERMICA SYSBAS to VIF-export
  - THERMICA .VIF
  - thermica_VIF_reader
  - thermica_SYSBAS_reader
  - thermica_SYSBAS_writer
  - temporary route via .VIF for quick development and verification
  - thermica_VIF_writer
  - esarad_erg_reader
  - esarad_erg_writer
  - STEP-TAS Repository Handler (produced by pyExpress)

- **Phase 2**
  - TRASYS_reader
  - TRASYS_writer
  - Part_21_reader
  - Part_21_writer
  - STEP-TAS Part.21 .step

**Development of pyExpress**

- Provide a STEP converter development environment, that
  - can be used by converter developer with minimal STEP knowledge
  - can be used as a Rapid Application Development tool for prototyping and near-real time validation and refinement of application protocols
  - has very strong string manipulation capabilities
  - maps well onto EXPRESS object oriented data models
  - leads to industrially robust converters, with acceptable performance and memory requirements, also for large models

- pyExpress is a STEP/EXPRESS compiler / code generator
  - Generates Python class library for implementation of converter in Python
  - Generates C++ class library for implementation of converter in C++
**pyExpress architecture**

- EXPRESS schema A
- Scanner / Tokenizer
- Parser
- SPARK
- pyExpress uses
- Python Code Generator
- C++ Generator
- Express MetaModel Handler
- pyExpress util

**Outline of converter based on pyExpress**

- Converter X-A-Y
- File or in-memory data for Tool X (input)
- Tool X to Schema A reader
- Schema A to Tool Y writer (To be produced by converter developer)
- Generated class library for Schema A
- Python or C++ API for Schema A
- Support library for Schema A: macro & convenience functions
- Generic STEP repository handler
- 'Schema A Library provided by pyExpress'
- Contains in-memory dataset(s) conforming to Schema A

- Output or input
- ISO 10303-21 file for dataset conforming to Schema A
- Part 21 writer
- Part 21 reader
- File or in-memory data for Tool Y (output)
**EXPRESS → Python Example (1)**

```python
class TAS Kepler parameter set(TasEntity):
    _atts_list_ =
    [sem1_major_axis, 'HSSF length measure',
     eccentricity, REAL,
     inclination, 'HSSF plane angle measure',
     right ascension of ascending node, 'HSSF plane angle measure',
     argument of perigee, 'HSSF plane angle measure',
     true anomaly at start, 'HSSF plane angle measure',
     _attr_id_namespace_]

def __init__(self):
    sem1_major_axis = 0.0
    eccentricity = 0.0
    inclination = 180.0
    right_ascension_of_ascending_node = 360.0
    argument_of_perigee = 90.0
    true_anomaly_at_start = 360.0

    # Validation
    assert _isinstance(HSSF length measure, sem1_major_axis)
    assert _isinstance(REAL, eccentricity)
    assert _isinstance(HSSF plane angle measure, inclination)
    assert _isinstance(HSSF plane angle measure, right_ascension_of_ascending_node)
    assert _isinstance(HSSF plane angle measure, argument_of_perigee)
    assert _isinstance(HSSF plane angle measure, true_anomaly_at_start)

    if sem1_major_axis > sem1_major_axis:
        sem1_major_axis = sem1_major_axis
    if inclination > inclination:
        inclination = inclination
    if right_ascension_of_ascending_node > right_ascension_of_ascending_node:
        right_ascension_of_ascending_node = right_ascension_of_ascending_node
    if argument_of_perigee > argument_of_perigee:
        argument_of_perigee = argument_of_perigee
    if true_anomaly_at_start > true_anomaly_at_start:
        true_anomaly_at_start = true_anomaly_at_start

    def set_sem1_major_axis(self, sem1_major_axis):
        assert _isinstance(HSSF length measure, sem1_major_axis)
        sem1_major_axis = sem1_major_axis
        sem1_major_axis = sem1_major_axis
        sem1_major_axis = sem1_major_axis

    def get_sem1_major_axis(self):
        return self.sem1_major_axis

    def __str__(self):
        return self.sem1_major_axis
```

---

**PyEXPRESS automatic generation**

- Complete Open Source code of new STEP-TAS/NRF C++ API, including STEP-21 write/read
  - automatically generated from EXPRESS schema
- Validated on:
  - Windows NT4/2000
  - Silicon Graphics /Irix 6
  - SUN/Solaris 8
**Updates to NRF and TAS protocols (1)**

- Removed AIM and mapping table, focussed on robust ARM
  - Interoperability with AP203 was not achieved in practice
  - Better do an executable AP203/TAS converter (retained mapping table for this)
  - Cost of full AIM implementation, verification and maintenance too high
- Major clean-up and replacement of unclear terminology
  - Resolved many issues collected over the years
  - Includes artificial constructs in ARM from original AIM/GR mapping
- Changed TAS navigational structure from bottom-up to top-down
  - Much easier / more natural to use in OO repository API
- Revalidated relationships for TAS geometry, meshing, thermal-radiative faces and made data model more consistent
Updates to NRF and TAS protocols (2)

• Updated NRF definitions for ‘datacube’, quantities and properties
  – Permitted all permutations of ordering (item, quantity, state)
  – Added a dedicated datacube for material properties
• Started to move protocol documents from MS Word to XHTML
• Revalidated all WHERE and RULE constraints
• Made all INVERSE attribute definitions consistent
• Consequence is that new STEP-TAS (ARM) Part 21 files are not compatible with previous STEP-TAS (AIM) Part 21 files
  – Not a serious problem since STEP-TAS was not yet really in industrial use
  – Last chance for this kind of updates

Current TASverter status

• Started in October 2002; 4th release made per 10 Oct 2003
  – Self-contained Windows, Solaris, Linux, Irix executables
    • No need to install Python
  – Free download from http://www.estec.esa.int/thermal/tools/tasverter.html
  – THERMICA .VIF and .SYSBAS readers/writers
  – ESARAD .erg reader/writer
  – STEP-TAS Part 21 reader/writer
  – Configuration controlled testsuite with unit and large model testcases, including fully automated run scripts for verification and regression testing
• CIGAL-2 reader/writer in progress
  – By Alcatel Space + OpenCascade (with some assistance from ESTEC)
**TASverter example 1: ISS_cold**

**THERMICA to ESARAD**

711 thermal-radiative surfaces, converted in less than 15 seconds.

---

**TASverter example 2: ATV**

**THERMICA to ESARAD**

1700 thermal-radiative surfaces, converted in less than 25 seconds.

Model hierarchy and coordinate transformations fully retained.
Verification Test Suite

- More than 200 unit tests
  - Documented as a website
  - with naming convention for subdirectories per testcase
  - actual and reference results for regression testing
  - Fully scripted to run and be diff-ed automatically

- Real model tests
  - ATV model
  - METOP C/D full spacecraft model
  - ISS model
  - Herschel and Planck full spacecraft models
Short term priorities

- STEP-TAS Verification Tool (in progress)
  - Semi-automatic verification of STEP-TAS Part 21 files
  - 3D visualisation for visual inspection
  - Extraction of key characteristics of exchanged model
    - Total surface area, surface area per aggregation level
    - Overall geometric envelop
    - Materials and material properties
    - Number of faces, surfaces
    - Number of thermal lumped parameter nodes, node ranges
  - Goal is to enable ESA to verify (certify) correctness of STEP-TAS datasets produced by different converters and to isolate cause of possible errors
- Up-to-date STEP-NRF and STEP-TAS documentation (in progress)
  - Including converter implementation examples

Outlook (short term)

- Submit STEP-NRF and STEP-TAS protocols to ISO TC 184 / SC 4 for ballot as ISO PAS or TS
- Publish schemas and tools/toolkits in open source (harmonisation)
  - pyExpress and TASverter
  - develop full capability pyExpress with University of Manchester
  - STEP-TAS and STEP-NRF schemas, Python and C++ libraries
- Provide and verify pyExpress generated C++ libraries for STEP-TAS
  - If requested by thermal analysis tool vendors
- ESA development STEP-SPE (Space Environmental Analysis)
  - Contract awarded, real work starting 27-Oct-2003
  - Extends TAS for micro-meteorites/debris, contamination, atomox, radiation …
Outlook (longer term)

- Promote implementation STEP-TAS in US, Canadian tools
  - TMG, Thermal Desktop, TSS, ...
- Possibly extend TASverter with new reader/writer plug-ins
  - Transform existing TRASYS/ESARAD converter to TRASYS reader/writer
  - Transform existing SINDA85/ESATAN converter to SINDA85 reader and ESATAN writer
  - Add more SINDA/ESATAN-like readers / writers
  - Add AP203 reader/writer, with primitive shape recognition capability
    - Can be derived from existing AP203/ESARAD converter plus old TAS AIM mapping; possibly add faceting of remaining NURBS surfaces
  - Construct HDF5 mapping and libraries for STEP-NRF

Related: STEP in the Building Industry

- Characteristic of Building Industry
  - A lot of independent large constructors and SME involved in one (big) building
  - Building becoming more and more complex
    - Structural, thermal, acoustic, electricity…
    - Hard regulation
  - CAD: Very large objects models
- STEP for building = IFC (Industry Foundation Classes)
  - Based on EXPRESS and STEP-21
  - Building dedicated integrated model:
    - architecture, materials, structural, thermal, HVAC…
- SIMULOG partnership, for STEP, CAD, Post-pro with GRAITEC
  - 3rd European software editor in the building industry
    - +3500 customers
    - +8000 licences CAD/analysis tools
    - +100 000 buildings designed
**PyExpress IFC-API (C++) used in 2 major projects**

- Building Design Chain (CSTB, Mediaconstruct)
  - Architects, designers \(\rightarrow\) CAD,
  - Engineers \(\rightarrow\) Structural Analysis
  - Engineers \(\rightarrow\) Thermal Analysis, Acoustic...
  - On-site workers \(\rightarrow\) measurement
  - Synthesis in virtual reality immersion room

- IFC-BRIDGE (SETRA)
  - IFC-BRIDGE (SETRA)
  - VINCI
  - setec
  - SNCF

---

**Acknowledgements**

- ESTEC team: Hans Peter de Koning, Simon Appel, David Alsina
- Contractor Simulog (France): Olivier Pailles, Arnaud Klinger
- Subcontractor GRAITEC (France): Eric Lebègue and co-workers
- ESARAD / ALSTOM Power (UK): Julian Thomas, David Scurrah
- THERMICA / Astrium SAS (France): Marc Jacquiau