Appendix X

Thermal Model Verification Guidelines
Draft Proposal

TEC-MTV
(ESA/ESTEC, The Netherlands)

Abstract

The use of computational analysis to support the development of S/C Thermal Control Systems (TCSs) is ubiquitous in modern industry. Thermal Models (TMs) are used during all phases of the S/C development and to support a large array of activities ranging from conceptual design right through to final in-flight predictions. Indeed, in some cases, thermal analysis is the only way that certain TCS requirements can be verified as physical tests are either too expensive or unrealisable. Because of this dependence upon computational analysis it is vital that there is a consistent approach to TM Verification and Validation (V&V). Ultimately such a V&V approach should improve the credibility of the predictions made using TMs.

The theme of V&V is well known in the context of quality assurance and systems engineering (including software systems). There has also been some work in other domains such as Computational Fluid Dynamics (CFD) and structural mechanics to develop processes for V&V of simulation models. In this particular context the following formal definitions usually apply:

- Verification is the process of determining that a computational model accurately represents the underlying mathematical model and its solution
- Validation is the process of determining the degree to which a computational model is an accurate representation of the real world from the perspective of the intended uses of the model

More informally the following questions, analogous with systems engineering, are often used:

- Verification "did we solve the equations correctly?"
- Validation "did we solve the correct equations?"

Whilst these definitions may be over simplistic they do allow the basic concepts of thermal model V&V to be communicated in just two short sentences.

ESA UNCLASSIFIED – For Official Use

esa
ESTEC

European Space Researchand Technology CentreKeplerlaan 12201
AZ NoordwijkThe Netherlands
T +31 (0)71 565 6565F +31 (0)71 565 6040www.esa.int

DOCUMENT

Thermal Model Verification Guidelines

DRAFT VERSION

Your suggestions & comments are welcome.

Prepared by	**TEC-MTV**
Reference	**TEC-MTV/2011/3562/ln**
Issue	**DRAFT**
Revision	**00**
Date of Issue	**24/10/2011**
Status	**Draft**
Document Type	**OTHER**
Distribution	

European Space Agency
Agence spatiale européenne

ESA UNCLASSIFIED – For Official Use

APPROVAL

Title	
Issue DRAFT	**Revision** 00
Author	**Date** 24/10/2011
Approved by	**Date**

CHANGE LOG

Reason for change	Issue	Revision	Date

CHANGE RECORD

Issue DRAFT	**Revision** 00		
Reason for change	**Date**	**Pages**	**Paragraph(s)**

ESA UNCLASSIFIED – For Official Use

Table of contents:

Page 3/28
estec
Date 24/10/2011 Issue DRAFT Rev 00

European Space Agency
Agence spatiale européenne

1 INTRODUCTION

1.1 Context

The use of computational analysis to support the development of S/C Thermal Control Systems (TCSs) is ubiquitous in modern industry. Thermal Models (TMs) are used during all phases of the S/C development and to support a large array of activities ranging from conceptual design right through to final in-flight predictions. Indeed, in some cases, thermal analysis is the only way that certain TCS requirements can be verified as physical tests are either too expensive or unrealisable. Because of this dependence upon computational analysis it is vital that there is a consistent approach to TM Verification and Validation (V&V). Ultimately such a V&V approach should improve the credibility of the predictions made using TMs.

The theme of V&V is well known in the context of quality assurance and systems engineering (including software systems). There has also been some work in other domains such as Computational Fluid Dynamics (CFD) and structural mechanics to develop processes for V&V of simulation models. In this particular context the following formal definitions usually apply:

- Verification is the process of determining that a computational model accurately represents the underlying mathematical model and its solution
- Validation is the process of determining the degree to which a computational model is an accurate representation of the real world from the perspective of the intended uses of the model

More informally the following questions, analogous with systems engineering, are often used:

- Verification “did we solve the equations correctly?”
- Validation “did we solve the correct equations?”

Whilst these definitions may be over simplistic they do allow the basic concepts of thermal model V&V to be communicated in just two short sentences.

1.2 Scope

The scope of the proposed document is limited to verification and the topic of validation will only be briefly touched upon. This is because the topic of validation is intrinsically linked to the topic of testing. Moreover, in a classical V&V process for computational models the task of verification comes before validation. It thus seems natural to address first verification, and to obtain feedback from users, before moving on to the topic of validation.

The intended users of the document are any persons, working in the domain of space systems, who use thermal analysis as part of their work. These users could be in industry, in agencies such as ESA or CNES, or in academia. Moreover, the guidelines should be applicable to users working on products at every level of the S/C product tree – that is to say at system level, sub-system level, unit level etc. The scope of the document (at least in early versions) will, however, be limited to “classical” S/C thermal analysis. This means that certain specialised topics will not be covered directly. Examples of these specialised topics might be re-entry systems, simulation of fluid loops and CFD for conjugate heat transfer.

Models are built at different levels (detailed dedicated model at unit/subsystem level) and have to be reduced for delivery and assembly to build the system level model. Tight planning leads to more and more automatization and few time-consuming analytical checks are performed. It is therefore crucial to define relevant checks and verification steps to ensure the validity of the model reduction, format change if any, delivery and correct assembly. This is necessary to validate the results obtained, and optimize the system tests and their correlation which are usually on a very critical path for planning.

Guidelines for those methodologies are necessary to ensure quality and acceptability of the approach at all levels (between companies and with agencies).

It is proposed that the style of the document should be practical in nature and the guidelines should be amenable to direct implementation by the users. The rationale for this is that most of the existing documents that address model V&V focus more on philosophy and processes than upon practical guidelines. Therefore, users who have an interest at this conceptual level already have a number of relevant sources to draw upon. The aim of producing "practical guidelines" is challenging, however, such a document has the best chance of being used.

1.3 Glossary

CDR	Critical Design Review
CFD	Computational Fluid Dynamics
CNES	French National Space Agency
COTS	Commercial Off-The-Shelf
CSG	ratio of nodal capacitance to sum of conductances
ESA	European Space Agency
FE	Finite Element
FEM	Finite Element Model
GMM	Geometric Mathematical Model
PDR	Preliminary Design Review
S/C	Space Craft
TCS	Thermal Control System
TM	Thermal Model
TMM	Thermal Mathematical Model
V&V	Verification & Validation

2 ADMINISTRATION OF THERMAL MODELS

2.1 Conventions

2.1.1 Language

2.1.2 Units

2.1.3 Coordinate System

2.2 Standardisation

2.2.1 Naming Conventions

2.2.2 Common Symbols

2.3 Configuration Control

2.3.1 Guidelines

Guideline 1: All thermal models should be placed under configuration control. The configuration control environment shall support the following features:

- Tracking of model changes with informative remarks
- Comparison (differencing) between distinct version of the model in the repository
- Tagging of model releases at critical milestones (e.g. PDR, CDR)

Guideline 2: Results of all production runs should be traceable to a specific version of the model inside the configuration control repository.

Guideline 3: The TMM & GMM couples shall be consistently tracked in the configuration control environment

Most thermal models of spacecraft are under some form of version control. However, this is often textual headers at the top of analysis files and manual incrementing of version numbers in file names.

There are many COTS and OS configuration control environments available (e.g. subversion, Mercurial), particularly for software development. These environments can be directly applied to thermal model configuration control, especially for ASCII formats. Moreover, many binary formats for documentation are also supported (e.g. .doc, .pdf). The use of such configuration control tools should not be a burden and will actually improve the efficiency and productivity of the analysts. Moreover, the maintainability of models over a number of years is vastly improved via the use formal version control.

2.4 Style

2.4.1 Comments

Guideline 4: Comments shall be in the English language for all models produced under ESA contract.

Guideline 5: Comments shall not be used to alter model topology, boundary conditions or procedural behaviour. Such conditions shall be implemented via user logic or alternative skeleton files etc.

Guideline 6: All user variables in a model shall be commented. The comments shall include:

- A short description of the data stored with the variable and intended purpose of the variable

- Identification of units where the variable relates to a physical quantity
- Justification of major thermal assumptions should be commented where it improves understanding of the model
- Readability of flow control structures (if …else … / select … case) & loop structures may also benefit from an adequate commenting of their purpose.

In 2.4.1, the use of comments in this way reduces the readability and maintainability of models. Such conditions are easy to overlook and shall be avoided. An illustrative example is shown in ESATAN syntax in snippet below.

```
# JRME 2011-10-12, Antenna hold-down conductors. Comment out these # conductors
for deployed cases

GL(1021, 3678) = 0.56;

GL(1022, 3686) = 0.56;
```

3 THERMAL MODEL CHECKS AND NUMERICAL VERIFICATION

3.1 Introduction

This section aims to cover the topic of thermal model checks and numerical verification, or, using the informal definitions introduced in TBD, "did we solve the equations correctly?"

It should be noted that typically the verification of computational models is split into ***code*** verification and ***calculation*** verification. Throughout the following discussion it is assumed that the code verification is carried out by the software vendors. Therefore, as users of the thermal analysis tools we need only concern ourselves with calculation verification.

3.2 Guidelines

3.2.1 Topology Checks

Many problems with thermal models can be attributed to ill-defined node/conductor topology in the model. As a minimum the following guidelines should be adhered to.

Guideline 7: Isolated nodes should be justified

Guideline 8: Conductively isolated groups of nodes should be justified

Guideline 9: Parallel conductors should be justified

Guideline 10: Negative or null conductors should be justified

Guideline 11: Negative or null nodal thermal capacities should be justified

3.2.2 Steady State Convergence

The adequate convergence of steady state analyses is a critical factor in ensuring the credibility of the model predictions. Unfortunately, and especially for large models, the computational time required to achieve adequate convergence can be significant. The temptation is thus to relax the convergence requirements in order to reduce computation time.

Guideline 12: The sensitivity of relevant model outputs to convergence criteria should be evaluated and appropriate limits agreed upon for the model. The following criteria shall be evaluated:

- Primary convergence criteria for iterative solutions (e.g. RELXCA/INBNDM in ESATAN)
- Energy balance (e.g. INBALA/INBALR in ESATAN)

> Guideline 13: Steady state production runs should be converged in the sense that all criteria listed in 3.2.2 are within the limits agreed with the customer.

In the guidelines above it is proposed that the term ***relevant outputs*** may be temperatures, heat flows, heater powers or any other pertinent model variables. Essentially, in a well converged model, the results that the user is interested in should be independent of any further tightening of the convergence criteria.
In reality the actual value of the convergence criteria will be highly model dependent and therefore hard numerical guidelines cannot easily be established. For example, the appropriate convergence criteria for a telecommunications platform model and a cryogenic instrument may be entirely different.

3.2.3 Transient Analysis

The use of transient thermal analysis to produce flight temperature predictions for the spacecraft is standard. However, the transient analysis, in the way it is used by thermal engineers, is also quite different from the types of analysis carried out in other computational domains. For example, a low-earth orbit may have a period of 100 minutes. Therefore, the model must be run for several orbits in order to reach a quasi-stabilised condition. This calls for long transient analyses adding to the computational demands. Once again, therefore, the thermal engineer must balance the computational effort against the accuracy of the model predictions.
The following guidelines aim to improve the credibility of transient analysis predictions by ensuring the convergence and stability of the solution process.

> Guideline 14: For transient runs using explicit solvers the time step should be smaller than the CSG limit

> Guideline 15: For transient runs using non-explicit solvers the time step should be smaller than half the shortest time constant in the model

> Guideline 16: The sensitivity of model outputs to transient solver criteria should be evaluated and appropriate limits agreed upon for the model. The following criteria should be evaluated:
>
> - Primary convergence criteria for iterative solutions (e.g. RELXCA/INBNDM in ESATAN)
> - Transient time step

Similarly to steady state analysis, the term ***relevant outputs*** may be temperatures, heat flows, heater powers or any other pertinent model variables.
3.2.3 regarding the CSG limit is necessary to ensure the stability of explicit solvers. Whilst this is a well known requirement from the theory of transient solvers, the use of explicit solvers is not common for space thermal analysis. Therefore 3.2.3 and 3.2.3 are more important when using implicit and Crank–Nicolson type solvers. There is an intrinsic inter-relation between these two parameters and a balance shall be sought such that the truncation and convergence errors are minimised. Ideally the model outputs shall be independent of the transient solver criteria although, in practice, the objective will be to reduce these errors to acceptable levels.
Where the smallest time constants in the model are very short then it may be advantageous to use arithmetic nodes for the lowest capacity elements in the model in order to increase the time step. Alternatively, in some tools, the use of local sub-stepping is possible, whereby the items with small thermal capacities use a smaller time step than the rest of the model.
Beyond numerical convergence of the solution, there are also other points to consider regarding transient analysis.

Guideline 17: The transient time step should be chosen such that the effects of imposed loads and boundary conditions are adequately resolved.

Guideline 18: The time range over which model results are observed should be driven by the model dynamic behaviour (either induced by the environment variations or by the thermal control operation) or the simulated mission sequence.

Guideline 19: The cyclic convergence should be assessed between successive time ranges and based upon criteria agreed with the customer that may address temperature differences and heating budget stability.

For example, if the model is subject to a short pulse of imposed heat input, the time step should be small enough to resolve the resulting temperature changes in the model. It happens also that the time step choice may be driven by the active thermal control itself (e.g. PID controller working at higher frequency) or by the results acquisition rate required to justify the meeting of a requirement (e.g. stability over a short period of time).

Moreover, regarding cyclic solution routines, where the heater cyclic period is of the same order as the orbit (or repeats analysis period) then assessment of the heater duty and budgets can become difficult.

3.2.4 Finite Element Models

The introduction of finite element methods into the thermal analyst's toolbox will lead to some specific additional requirements. These requirements are quite generic for all finite element models across application domains. The actual safe limits used for topology check can probably be less restrictive for thermal models compared with, say, structural models i.e. we can probably use worse elements in thermal models. Nonetheless the following guidelines should be adhered to ensure the quality of finite element meshes.

Guideline 20: The geometrical adequacy of finite elements should be checked to be within the limits defined in TODO. The following criteria should be checked: warp, skew, interior angle, aspect ratio [ref]

Guideline 21: Duplicate or overlapping elements should be justified

Guideline 22: Duplicate finite element nodes should be justified

Guideline 23: The topological connectivity of finite element meshes should be checked using the following utilities:

- Free edges (for 2D and 3D elements)
- Free faces (for 3D elements)

3.2.5 Radiative Computations

Discussion TODO

Guideline 24: For models containing surfaces with non-zero specularity, an appropriate method should be used. Examples of appropriate methods are: MCRT

Guideline 25: For MCRT computations, the sensitivity of relevant model outputs to input parameters of the ray-tracing algorithm should be evaluated and appropriate limits agreed upon for the model:

- The sensitivity analysis should consider both radiative couplings and heat fluxes
- The sensitivity analysis should consider measures of statistical convergence such as line accuracy, reciprocity and variation of random number seeds
- The sensitivity analysis should consider end-to-end results from the thermal solution (e.g. temperature, heat flows etc.) due to ray-tracing parameters.

Guideline 26: The sensitivity of relevant model outputs to the filtering of radiative couplings should be evaluated and appropriate limits agreed upon for the model:

- The sensitivity analysis should consider end-to-end results from the thermal solution (e.g. temperature, heat flows etc.) due to ray-tracing parameters.

Guideline 27: For a given face, the REFs to inactive surfaces shall make up less than TODO of the total REFs from that face.

Guideline 28: The sensitivity of relevant model outputs to the number of orbital positions shall be evaluated and appropriate limits agreed upon for the model.

3.3Additional Guidelines

More points to be added TODO

Guideline 29: Tabulated data shall take make provision for “end-conditions.” Extrapolation outside of table bounds shall not be occur during the solution routine

4 GUIDELINES FOR CODING AND USER LOGIC

4.1 Introduction

If the thermal model and the structural FE model of a given spacecraft were to be compared, one of the most obvious differences would be the amount of procedural code, or user logic, to be found in the thermal model. This user logic, typically written in a language such as FORTRAN, is available in most of the thermal analysis tools for space applications (at least in ESATAN, THERMISOL and SINDA) and provides almost limitless flexibility to the user. This flexibility is tremendously valuable for many applications such as handling non-standard cases, modelling specific thermal control hardware or for customised reporting and data processing. However, along with this flexibility comes a certain amount of risk. There is always potential for programming errors to be introduced into user logic and even the most advanced pre-processor or syntax checker cannot guard against all of these errors.

Generally speaking the users of thermal analysis tools take a pragmatic approach to writing user logic – if the logic seems to have the desired thermodynamic effect on the model then it is probably OK. This is an entirely understandable view given that the users probably have very little formal training in software engineering (maybe an undergraduate course or two). However, the code that is written is often quite complex and represents a very significant amount of work. Moreover, the life time of the generated models can be many years (the full S/C development plus possible operational usage) and during this period it is likely that several users will work on, modify or even just read the code. Therefore the introduction of some coding guidelines is a key factor in improving the quality and maintainability of thermal models.

The following guidelines are a mixture of some standard FORTRAN-like coding conventions (many of which can be found online) and some thermal modelling specific points. The guidelines are strongly driven by the input formats of the standard tools for space thermal analysis in Europe notably the ESATAN syntax, however, they may also generally applicable to other tools such as user subroutines in TMG or NASTRAN.

4.2 Guidelines

4.2.1 Minimising the Number of Warning Messages

As a general rule the user should try to minimise the number of warnings generated by the analysis tool. This may seem like an obvious statement, however, experience shows that many models generate a lot of warning messages; often for trivial syntactical inconsistencies. The problem is that, whilst these warnings may not adversely affect the analysis results, they can mask other more significant warnings which the user should take note of.
In order to reduce the number of warning messages, the following guidelines should be adhered to.

> Guideline 30: Each auxiliary variable with scope limited to a single operations block should be declared at the start of that operation block

Very often the user wishes to create an auxiliary variable within an operations block. Often this variable is only used within the scope of that block, for example; a common example of this would be counter variables used in a do-loop. If these variables are not declared then the tool may generate warning messages in the log file.

To reduce the number of these warning messages all auxiliary variables should be declared at the top of the operations block in which they are used. Note that in FORTRAN 77 it is required that all variables are declared at the beginning of a subroutine. In ESATAN all operations block as are mapped to subroutines by the pre-processor and therefore variables should be declared at the top of the operations block.

Guideline 31: Only flow control structures that pass through the pre-processor without generating warnings should be used

The use of flow control structures such as do-loops in operations blocks is widespread, however, it is observed that associated warning messages are often produced. This is because the pre-processor expects only FORTRAN-77 style loops of the form:

```
      DO 100, ICOUNT 1, 10

         ......

100   CONTINUE
```

Often, however, FORTRAN-90 style loops are used of the form:

```
      DO ICOUNT 1, 10

         ......

      END DO
```

These loops will create warnings, although they may pass through the compiler and execute correctly.
To reduce the number of warning messages the user should ensure that all do-loops are of the FORTRAN-77 style and are terminates by a separate continue. Alternatively, if this entails too much effort, other flow control structures such as REPEAT, UNTIL or WHILE, ENDWHILE may be considered.

4.2.2 Coding Style

Guideline 32: The use of tab characters to generate whitespace in user generated code shall be avoided. Spaces should be used in place of tabs.

To improve portability of the user generated code, both across platforms and between tool chains, it is recommend to use spaces, rather than tabs, to implement whitespace.
The use of tabs to generate whitespace can mean that the formatting of the file is not preserved when moving between platforms or tool chains (e.g. text editors). This affects the readability of the user generated code, especially if a mix of tabs and spaces have been used. In some cases the use of tab characters can also lead to syntax errors during the pre-processing of the model.

Guideline 33: The body of flow control structures should be indented.

The use of indentation in programming languages is an important concept which helps to convey the program flow and structure. Whilst indentation is not formally required in most programming languages (with notable exceptions such as Python) it is strongly recommended to improve the readability of the code. In particular the use of indentation helps to clearly and quickly identify flow control structures such as loops and conditions.

Just like in any other computer program, the use of indentation in thermal models can only help to improve the readability maintainability of the user generated code. The size of the indent is not essential, however, the use of a consistent indent level throughout the code is recommended.
The use of spaces is recommended to implement indentation rather than tab characters.

Guideline 34: Use subroutines and functions to logically structure user generated code.

Guideline 35: Keep subroutines and functions short

Good programming practice dictates that code should be split up into logical sections in functions or subroutines. If all the code is gathered in one main program or subroutine then it is difficult to have a full overview of what that code does on a single computer screen. The user must therefore scroll up and down the code leading to disorientation and potential loss of context.
As a general rule the code is clearer to understand if related things are kept close together. For example, one rule of thumb is that the contents of any flow control structure, function or subroutine should fit within one computer screen. To achieve this the user is forced to move large or repeated blocks of code to subroutines or functions. Moreover, if subroutines and functions are kept short then the declaration of variables at, the top of the subroutine, will be close to the location where they are used. This again helps with the readability and maintainability of the code.

NOTE 1. There are of course many examples of subroutines which are very long, e.g. auto-generated solar fluxes. This is not a problem because they are auto-generated and the user need not traverse them regularly.

4.2.3 *Variable Naming*

Compared with more modern programming languages older version of FORTRAN were restrictive in terms of the permitted naming for variables, for example they were limited to 6 characters. More recently this limitation has been relaxed in the analysis tools and variable names of up to TODO characters are permitted. The user should therefore take advantage of this increased variable name length in order to improve the readability and maintainability of the code.

Guideline 36: The user should aim to make variable names clearly readable

Guideline 37: Variable names should be in the English language for all models produced under ESA contract

The readability of user produced code is improved if the variable names can be clearly identified. In the past common practice was to use all uppercase variable naming, often limited to only 6 characters. Better readability can be achieved using, for example, mixed case naming of the form:

```
INTEGER*loadCase = 1;              # [-] 1 for hot case

                                   #     2 for cold case



REAL*detectDissip = 60.0D-3;       # [W] Detector dissipation
```

The exact naming convention used is not as important as giving thought to this issue and maintaining consistency throughout the model.

Guideline 38: A variable name should give an indication of the physical quantity stored within it.

The readability of user produced code is improved vastly if the naming of variables or user constants gives an indication of the physical quantity is represents. For example, a some examples using one such convention ar shown below:

```
REAL*thkPanel = 0.45D0;        # [m] Thickness of panel
REAL*kAl6061 = 185.0D0;        # [W/mK] Conductivity of Aluminium 6061
INTEGER*numBolts = 5;          # [-] Number of bolts around flange
REAL*condBolt = 0.05;          # [W/K] Total conductance of bolted
                               #       interface including washers
```

Such a naming convention helps to improve the readability of the code and, moreover, increases the chance of detecting human errors of the form:

```
GL(300, 305) = condBolt + numBolts;
```

which are evidently dimensionally incorrect upon first inspection of the code.
It should be noted that the actual naming convention used is not as important as maintaining consistency throughout the model and across variables.

4.2.4 *Access to Solver Internal Variables*

Guideline 39: Internal variables and arrays of the solver should not be directly accessed or set by the user.

In tools such as ESATAN the internal data structure is often a series of arrays which can be indexed to obtain model entities. These data structures are, however, internal to the tool and do not form part of the public interface of the software. It is therefore risky to use these variables because they could change at any time, for example due to restructuring of the code by the developer.
Moreover, whilst use of these arrays may provide convenient shorthand, it relies upon a knowledge of the internal data structures which is often not available in the user documentation. Therefore for less experienced users the code is difficult to interpret complex to maintain over time.
An example of the use of internal variables to set the temperature of all nodes of a model except the last 2, which are in this case the inactive (99998) and space (99999) nodes, is shown below.

```
      DO, 100, ICOUNT 1, FLG(1)- 2
         T(ICOUNT) = 20.0D0
100   CONTINUE
```

This is convenient for syntax for experienced users, however, it relies on knowledge of the FLG array contents and the fact that there is a array of temperatures internally. A better solution which could be implemented (although not the only one) would be to use an ESATAN public routine, for example:

```
      CALL SETNDR('#1-99997', 'T', 20.0D0, CURRENT)
```

5 TRANSFER OF THERMAL MODELS

5.1 Introduction

The transfer of thermal models between parties is a task that occurs many times during the course of a typical space project. For example, models of equipment or subsystems are regularly provided by sub-contractors to customers for integration into a higher level model. Prime contractors also regularly provide system level models to customers (e.g. ESA) or reduced models to launch authorities for coupled analysis. Unfortunately, every time a model transfer occurs there is the potential for problems to arise.
Some examples of the kind of problems that can occur when exchanging models between parties are given in the following (non-exhaustive) list:

- Corruption, or even loss, of electronic data
- Incomplete or incorrect deliveries meaning that the model cannot be executed (e.g. missing files)
- Incomplete or inadequate documentation describing the model and how to execute it
- Portability problems such as the use of different operating systems (e.g. MS Windows, Linux, HP)
- Problems associated with supporting tools required to execute an analysis (e.g. proprietary, obsolete or in-house tools etc.)

The following guidelines aim to establish best practice for the transfer of thermal models between parties.

5.2 Guidelines

5.2.1 Required Analysis Files and Reference Results

The fundamental items in any model delivery are the analysis files themselves; usually both geometrical models and thermal models are included. For a formal delivery, associated with a project milestone, there are also typically a number of scenarios which are delivered relating to worst cases, different operation models, different configurations (e.g. stowed, deployed) etc.
In order to make the transfer of thermal models as seamless there is a minimum set of deliverable model files which are necessary.

> Guideline 40: A formal model transfer should contain all the necessary components to execute a complete analysis run.

When a thermal model is transferred between parties, the recipient should be able to directly execute a complete analysis run and obtain results. In order for this to be possible it is essential that the delivery contains all of the necessary components to execute an end-to-end analysis. Here the term components may refer to:

- All of the ***analysis files*** together with associated include files and global files
- Any ***external libraries*** or routines required to run the model. For example externally linked FORTRAN routines for material properties or results processing
- Any ***supporting tools*** such as run scripts, or EXCEL based tools, which are used to execute the analysis chain. For example tools used to: extract radiative couplings or fluxes, set up analysis cases, create results directories, or carry out other pre- and post-processing

Guideline 41: A formal model transfer should contain, for each analysis case provided, a set of reference results to be used for verification of the delivery. Reference results should be in raw data files in the same format as produced by the analysis process.

Guideline 42: The execution of the analysis cases provided should yield identical results to those provided with the delivery

Assuming that a complete set of analysis files is provided in-line with , the recipient should be able to directly execute the model and obtain results. The results can then be compared to those provided in the delivery. The purpose of this comparison is to ensure that the delivered files were not corrupted in any way, and that the recipient's tool-chain is capable of producing results consistent with the supplier's.
In principle the recipient's results should be numerically identical to the reference results, although some differences may be expected due to different computing architectures (32 or 64 bit) or different versions of the analysis software. For example, enhancements or bug fixes in the analysis software may lead to numerical differences. Generally speaking, however, these kind of numerical differences should be several orders of magnitude (TODO) lower than the uncertainty applied to the analysis predictions.

5.2.2 Documentation

The formal transfer of thermal models should be accompanied by supporting documentation that allows the recipient to install and use the models on their computing system. This may be a standalone document, a read-me file, or it may form part of the thermal model description document (see ECSS []). Nonetheless it is an essential part of any model delivery.

Guideline 43: The documentation provided with a formal model transfer should contain full end-to-end instructions on how to install and run the delivered analysis cases. This should also include:

- Description and usage of any software utilities, in addition to the thermal analysis tools, required to run the analysis cases
- Description of any manual steps that are required to run the analysis cases

Guideline 44: The documentation provided with a formal model transfer should contain the following administrative information:

- Versions of all thermal analysis software used to produce reference results
- Versions of all thermal models in the supplier's configuration control environment
- Computational architecture and platform used by the supplier and used to generate the reference results

The provision of the information described in the previous guidelines is essential in order for the recipient to be able to execute the model with minimum effort. Moreover it is important to establish a traceable workflow from the model files to the reference results. This is especially important when the long lifetime of space projects, and the number of people who may work on a given project, is considered.

In addition to the formal documentation discussed in the guidelines above there are, from experience, many other more subtle points that can cause difficulties during model transfer. Whilst these points are difficult to formalise in guidelines, several such issues are listed in the bullet points below:

- Most thermal model transfers use an electronic archive of some sort (e.g. zip or tar). It should be noted that this can have unforeseen consequences such as loss of model directory structure and loss of symbolic links used to organise model files.

- Often the thermal models delivered contain some sort of hard-codes file paths which can cause problems on the recipients file system. If the models need to be unpacked in a specific directory structure, or if certain file paths are required, then this should be flagged in the delivery documentation

5.2.3 Portability of Thermal Models

In order to improve the portability of thermal models between computing platforms (e.g. between Windows and Linux) the following guidelines are proposed:

> Guideline 45: Limit file and directory names to the characters A-Z, a-z, 0-9, full stop, hyphen, and underscore.

> Guideline 46: Do not use full stop in directory names.

When software utilities, additional to the thermal analysis tools, are required to execute a full analysis run, then consideration should be given to the portability of the tools. For example if the extraction of external heat fluxes, and processing for input to the TMM, is carried out using a Visual Basic program then it will be difficult to execute the complete workflow on a Linux system. The same concern is applicable to in-house tools which cannot be distributed.

> Guideline 47: Supporting software utilities should be portable across computing platforms.

> Guideline 48: Supporting software utilities should not be based on proprietary software which cannot be included in a thermal model delivery

6 GUIDELINES FOR MODEL CONVERSION

6.1 Introduction

6.2 Guidelines

TODO

- System subroutines, e.g. Thermostats on/off variable inversed
- Defaults orbit parameters can be different – small g
- Arithmetic nodes – SINDA/ESATAN
- Double side inactive shells (blocker, invis.)
- Variable naming length (SINDA limit)
- Realistic test cases that actually test logic – e.g. heaters
- Units
- Nodal quantities

7 CRITICAL FEATURES, PITFALLS & TIPS IN THERMAL MODELLING

NOTE 1. This prototype chapter is to offer an alternative or complementary way of presenting verification items by addressing them directly in users' reference frame, i.e. the model input file structure.

7.1 Thermal mathematical models (TMM)

This chapter lists the critical features that need to be questioned as one performs a thermal model assessment. It parallel addresses the most common pitfalls and provides a number of good-practice considerations that ESTEC would like to foster in order to ease the exchange process of thermal models within the community.

This discussion should be regarded as a top-level verification guideline and is not intended to supersede any of the different user manuals provided by thermal software editors.

Most common thermal network analysers (such as ESATAN, SINDA or THERMISOL for instance) share, with some nuances, a similar card-structured syntax, as far as their input files are concerned. That's the reason why it has been deemed appropriate in practice to sort the different discussed items according to the ESATAN-like card they belong to.

7.1.1 $MODEL

This section is appropriate to gather configuration information (Cf. paragraph 2.3.1).

2.3.1 All thermal models should be placed under configuration control. The configuration control environment shall support the following features:

2.3.1 Results of all production runs should be traceable to a specific version of the model inside the configuration control repository.

2.3.1 The TMM & GMM couples shall be consistently tracked in the configuration control environment

7.1.2 $LOCALS

Run speed-up opportunity

TODO

Standardization opportunity (Cf. paragraph 4.2.3)

4.2.3 The user should aim to make variable names clearly readable

4.2.3 Variable names should be in the English language for all models produced under ESA contract

4.2.3 A variable name should give an indication of the physical quantity stored within it.

Parameterization

7.1.3 $NODES

3.2.1 Isolated nodes should be justified

3.2.1 Negative or null nodal thermal capacities should be justified

Number of nodes

A proper thermal lumped network should conform to certain basic rules as far as spatial discretization is concerned.
For instance, the **isothermal assumption** that basically governs the thermal nodal breakdown shall be assessed with respect to the targeted accuracy and to the needed observables that shall justify the thermal design performances.

Temperature requirements generally apply to specific locations called **temperature reference points** (TRP). It is quite important to properly render those points in the thermal model breakdown in order to allow a straightforward comparison. There are other usual requirements (gradients, gradients stability, heating power …) that may require **local refinements** of the nodal breakdown to allow a proper assessment.

Automatic network generation routines show great interest, in terms of initial effort to get a thermal network namely, but sometimes provide so deeply involved and numerically intricate models that they may simply prohibit any further thermal analysis. A thermal model should allow to still comprehend the **physical phenomena** at stake (e.g. intuitive couplings, flux evolution). Marginally, the huge number of nodes generated may become also out-of-range for network analysers and post-processing tools capabilities.
Directly linked to the way the model is discretized, there is a real interest, numerically speaking, for the most common transient solution routines to avoid a great dispersion of the couplings values (typically a factor 1000 between maximum and minimum conductive couplings). Same recommendation stands for thermal capacities. This may otherwise disturb numerical convergence and drastically slow down the run completion.

Numbering philosophy

In the perspective of collaborative effort, specific numbering conventions might be used to ease sub-models reconciliation and integration inside the top-level model. The use of some functions or routines (e.g. heat flux functions) may be drastically facilitated if a methodical numbering is adopted.

Sink temperatures

Handle with care. TODO

Fluid modelling

TODO

Clear and explicit labelling required

TODO

Arithmetic nodes

TODO

Sub-models

TODO

7.1.4 $CONDUCTORS

Conductive couplings

3.2.1 Conductively isolated groups of nodes should be justified

3.2.1 Parallel conductors should be justified

3.2.1 Negative or null conductors should be justified

Automatic conductor generation (warnings)
Care for parameterization capabilities

Radiative couplings

2.3.1 The TMM & GMM couples shall be consistently tracked in the configuration control environment

Fluidic couplings

TODO

7.1.5 $CONSTANTS

2.4.1 All user variables in a model shall be commented. The comments shall include:

4.2.3 The user should aim to make variable names clearly readable

4.2.3 Variable names should be in the English language for all models produced under ESA contract

4.2.3 A variable name should give an indication of the physical quantity stored within it.

7.1.6 $CONTROL

Convergence criterion

Cf. paragraph 3.2.2 & 3.2.3.

3.2.2 The sensitivity of relevant model outputs to convergence criteria should be evaluated and appropriate limits agreed upon for the model. The following criteria shall be evaluated:

3.2.2 Steady state production runs should be converged in the sense that all criteria listed in 3.2.2 are within the limits agreed with the customer.

3.2.3 For transient runs using explicit solvers the time step should be smaller than the CSG limit

3.2.3 For transient runs using non-explicit solvers the time step should be smaller than half the shortest time constant in the model

3.2.3 The sensitivity of model outputs to transient solver criteria should be evaluated and appropriate limits agreed upon for the model. The following criteria should be evaluated:

3.2.3 The transient time step should be chosen such that the effects of imposed loads and boundary conditions are adequately resolved.

3.2.3 The time range over which model results are observed should be driven by the model dynamic behaviour (either induced by the environment variations or by the thermal control operation) or the simulated mission sequence.

3.2.3 The cyclic convergence should be assessed between successive time ranges and based upon criteria agreed with the customer that may address temperature differences and heating budget stability.

7.1.7 $ARRAYS

3.3 Tabulated data shall take make provision for "end-conditions." Extrapolation outside of table bounds shall not be occur during the solution routine

Temperature dependent items

According to the system sensitivity to this topic and in particular when dealing with cryogenic temperatures, the temperature dependence of materials properties (e.g. thermal conductivity or capacitance) shall be properly addressed.
A few thermal hardware products require an explicit expression of their key parameter in function of temperature (e.g. louvers opening angle, Peltier device cooling efficiency).

Time dependent items

- Mission timeline
    - Phases sequence (e.g. electronics dissipation)
    - Eclipse flag
    - Aerothermal flux
    - Altitude (marginally)
- External fluxes
    - Solar radiation
    - Albedo
    - Infrared radiation
- Interfaces
    - Sink temperatures
    - Interface fluxes

Mission control

- Mode selection
- Supply voltage

7.1.8 $SUBROUTINES

Cf. paragraph 4.2

4.2.2 The use of tab characters to generate whitespace in user generated code shall be avoided. Spaces should be used in place of tabs.

4.2.2 The body of flow control structures should be indented.

4.2.2 Use subroutines and functions to logically structure user generated code.

4.2.2 Keep subroutines and functions short

4.2.4 Internal variables and arrays of the solver should not be directly accessed or set by the user.

7.1.9 $INITIAL

2.4.1 Comments shall not be used to alter model topology, boundary conditions or procedural behaviour. Such conditions shall be implemented via user logic or alternative skeleton files etc.

4.2.1 Each auxiliary variable with scope limited to a single operations block should be declared at the start of that operation block

4.2.1 Only flow control structures that pass through the pre-processor without generating warnings should be used

7.1.10 $VARIABLES1

2.4.1 Comments shall not be used to alter model topology, boundary conditions or procedural behaviour. Such conditions shall be implemented via user logic or alternative skeleton files etc.

7.1.11 $VARIABLES2

2.4.1 Comments shall not be used to alter model topology, boundary conditions or procedural behaviour. Such conditions shall be implemented via user logic or alternative skeleton files etc.

7.1.12 $EXECUTION

2.4.1 Comments shall not be used to alter model topology, boundary conditions or procedural behaviour. Such conditions shall be implemented via user logic or alternative skeleton files etc.

Appropriate routine

Starting point

- Steady-state routine
  - Requires mean conditions
- Initialization file
  - Risk of loss of status constants

Model consistency check

TODO

7.1.13 $OUTPUTS

Heat flux

Curves

Tables

- Gradients
- Min/ave/max

7.2 Radiative models

Cf. paragraph 3.2.5.

3.2.5 For models containing surfaces with non-zero specularity, an appropriate method should be used. Examples of appropriate methods are: MCRT

3.2.5 For MCRT computations, the sensitivity of relevant model outputs to input parameters of the ray-tracing algorithm should be evaluated and appropriate limits agreed upon for the model:

3.2.5 The sensitivity of relevant model outputs to the filtering of radiative couplings should be evaluated and appropriate limits agreed upon for the model:

3.2.5 For a given face, the REFs to inactive surfaces shall make up less than TODO of the total REFs from that face.

3.2.5 The sensitivity of relevant model outputs to the number of orbital positions shall be evaluated and appropriate limits agreed upon for the model.

Accuracy assessment

- Appropriate sized surfaces vs. number of rays
- Statistical error estimate
- Filtering of REFs
    - Percentage of lost energy
    - Not with space
- Special care when opticals are present
    - Analytical surfaces

Thermo-optical properties

- Robustness
    - Sources to be identified
    - Parameterised
- Main concerns
    - Low emissivity or absorptivity => increase the number of rays
    - Transmissivity
    - Wavelength dependence
    - Incidence angle dependence
        - UV/IR specularity
        - Non-lambertian coatings
    - Ageing factors
        - UV
        - Atomic oxygen
        - Radiation
    - Electrical conductivity

7.3 Conductive models

Physical properties

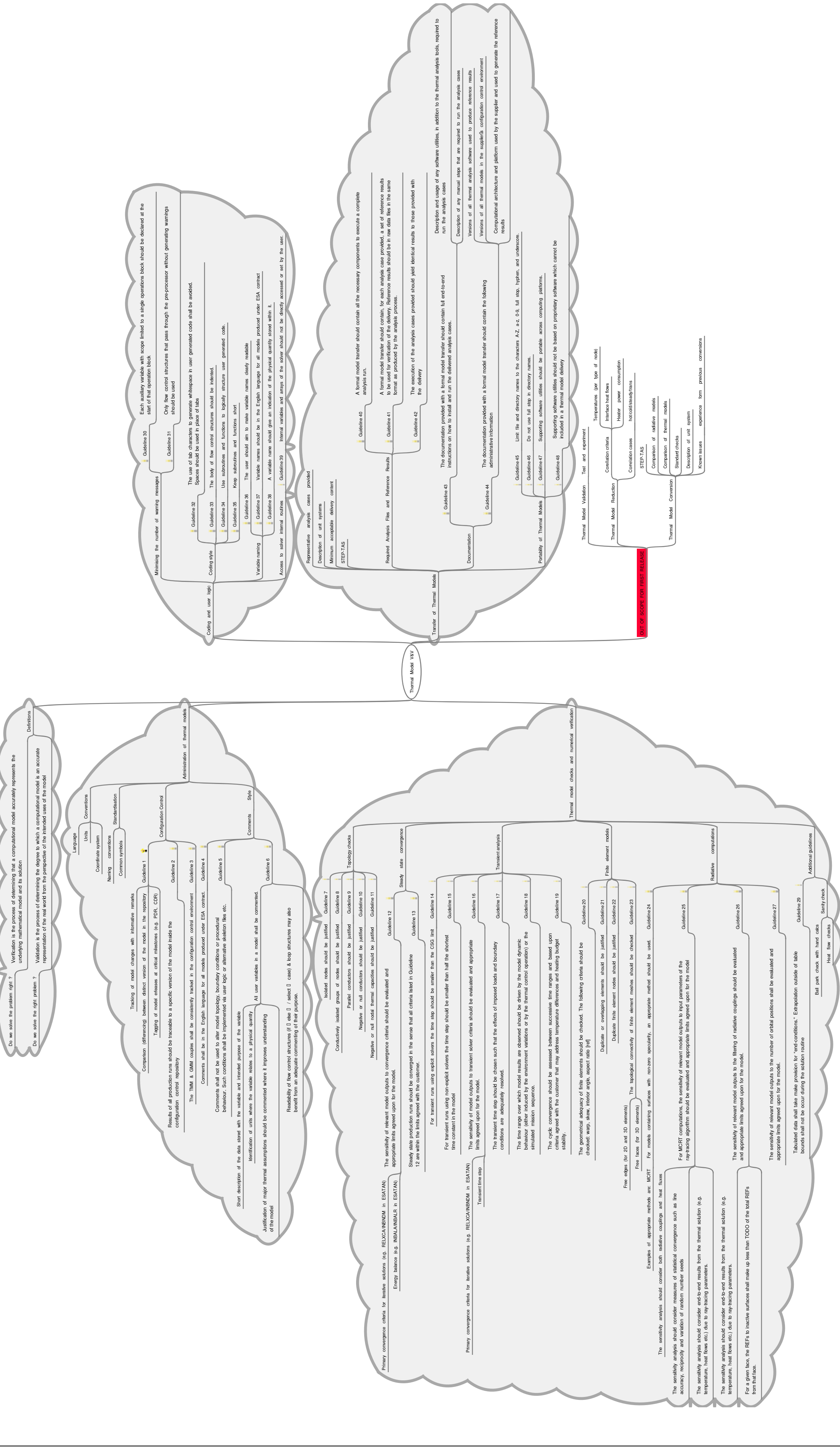
- Thermal conductivity
- Thermal Capacity

Thicknesses

- Parameterised

Interfaces

- Edge detection
- Edge behaviour


Thermal Model V&V
Definitions
Administration of thermal models
Thermal model checks and numerical verification
Coding and user logic
Transfer of Thermal Models
OUT OF SCOPE FOR FIRST RELEASE