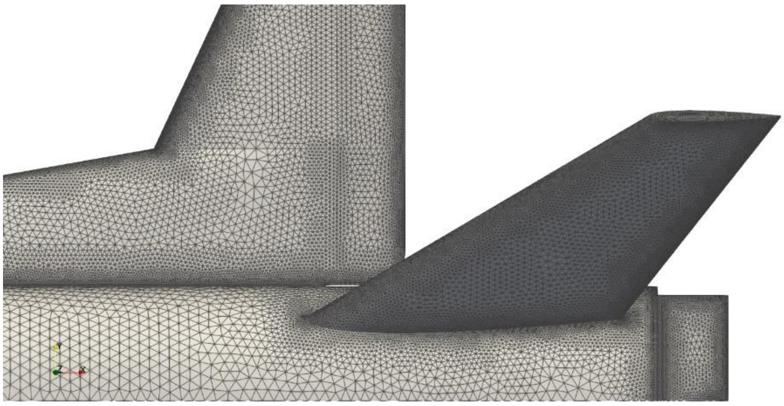
ATSI
EXECUTIVE SUMMARY

Aerothermodynamics Tools for System Integration



Code:	ATSI-DES-TEC-EXS
Issue:	1.0
Approval Date:	08/06/2023
Confidentiality Level:	Unclassified

Approval Signature:

Prepared by: ATSI Development Team
Reviewed by: P. Hermosín / Project Manager
Approved by: P. Hermosín / Project Manager

Digitally signed by Pablo Hermosin
Date: 2023.06.08 10:36:49 +02'00'

This page intentionally left blank

Document Status Log

Issue	Section	Change Description	Date
1.0	All	First release	08/06/2023

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1. Background

Throughout the lifetime of ESA missions, from conceptual design (Phase 0) until disposal (Phase F), an enormous amount of data is generated in each of the phases for the different engineering disciplines, e.g. aero(thermo)dynamic, structural and thermal analysis, models of the propulsion subsystem or design trajectories.

Assembly, Integration, and Testing (AIT) processes occur during the phases C and D of the engineering processes of a given flight vehicle. System integration processes take place at the level of engineering models, structural and thermal models, and qualification models of flight vehicles.

The current aerothermodynamics (ATD) tools cover the phases 0, A, and part of phase B for design, and development of flight vehicles. Beyond those phases, the current toolset is not able to cover tasks of system integration and testing (phases C, D). This is because the integration and testing flow of flight vehicles produces substantial amount of new data at each integration cycle that cannot be input and processed using the current ATD toolset at the required latency and quality: structure integration data, thermal data, propulsion integration data, configuration data, etc.

ATSI is a toolset devoted to support the integration of numerical system models, by means of individual modules to interface different numerical models, transferring and projecting data from one model to another, prepare and run the relevant engineering tools, which include at least the disciplines of CAD, aerodynamics, propulsion, trajectories, structures and thermal.

1.2. Purpose

The objective of this document is to provide the Executive Summary for the Aerothermodynamics tools for System integration (ATSI) activities, with a summary of the background, design and development, functionalities and outcome of the ATSI tool.

1.3. Scope

This document has been produced in the frame of the WP 6000 (Project Management) of the contract 4000135887/21/NL/CT between ESA-ESTEC and DEIMOS Engineering and Systems (DES) for the study on.

1.4. Acronyms and Abbreviations

The acronyms and abbreviations used in this document are the following ones:

AIT	Assembly, Integration and Testing
AEM	Attitude Ephemeris Message
ATD	AeroThermoDynamics
ATSI	Aerothermodynamics Tools for System Integration
CAD	Computer-Aided Design
CDS	Common Data Structure
CFD	Computational Fluid Dynamics
CI	Continuous Integration
CSV	Comma-Separated Values
DES	Deimos Engineering and Systems

DLR	Deutsches Zentrum für Luft- und Raumfahrt
DMS	Deimos Space
EAI	Empresarios Agrupados Internacional
ECSS	European Cooperation for Space Standardization
ESPSS	European Space Propulsion System Simulation
FEM	Finite Element Model
FSI	Fluid Structure Interaction
GUI	Graphical User Interface
IC	Initial Conditions
IGES	Initial Graphics Exchange Specification
OEM	Orbit Ephemerides Message
OML	Outer Mold Line
OS	Operating System
PFAT	Post-Flight Analysis Tool
RVM	Reference Vehicle Model
SoW	Statement of Work
STEP	Standard for the Exchange of Product model data
STL	Stereolithography
SUM	Software User Manual
SW	Software
UC	Use Case
WP	Work Package

1.5. Definitions

The definitions of the specific terms used in this document are the following ones:

- **Acceptance Testing**: Formal testing conducted to determine whether or not a system satisfies the acceptance criteria, previously defined by the customer.
- **Graphical User Interface**: interface that allows users to interact with the system through graphical icons and visual indicators.
- **Integration Testing**: An orderly progression of testing in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated.
- **System Testing**: The process of testing an integrated hardware and software system to verify that the system meets its software requirements.
- **Test Case**: Documentation specifying inputs, predicted results, and a set of execution conditions for a test item.
- **Test Plan**: Documentation specifying the scope, approach, resources, and schedule of intended testing activities.
- **Test Procedure**: Documentation specifying a sequence of actions for the execution of a test.
- **Use Case**: a list of action or event steps, typically defining the interactions between a role (known in the UML as an actor) and a system, to achieve a goal. The actor can be a human, an external system, or time.
- **Verification:** Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled [ISO 9000:2005]

NOTE: verification process (for software) is the process to confirm that adequate specifications and inputs exist for any activity, and that the outputs of the activities are correct and consistent with the specifications and input.

- **Validation:** Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled [ISO 9000:2005]

  NOTE: The validation process (for software) is the process to confirm that the requirements baseline functions and performances are correctly and completely implemented in the final product.

2. RELATED DOCUMENTS

2.1. Applicable Documents

The following table specifies the applicable documents that shall be complied with during project development.

Table 1: Applicable documents

Ref.	Code	Title	Issue
[AD 1]	AO/1-10736/21/NL/MG	Invitation to Tender AO/1-10736/21/NL/MG, "Aerothermodynamics Tools for System Integration"	10/03/21
[AD 2]	ATSI-DES-COM-PRS01-E	"Aerothermodynamics Tools for System Integration" Proposal	19/05/21
[AD 3]	ATSI-DES-TEC-TNO1.1-E	"Use case analysis and software requirements identification"	08/06/23
[AD 4]	ATSI-DES-TEC-TNO1.2-E	"Architecture Design Document"	08/06/23
[AD 5]	ATSI-DES-TEC-TNO3.1-E	"Validation and Testing Document"	08/06/23
[AD 6]	ATSI-DES-TEC-TNO3.2-E	"Software Manuals and Tutorials"	08/06/23
[AD 7]	ATSI-DES-TEC-TNO3.3-E	"Future Developments Roadmap"	08/06/23
[AD 8]	ATSI-DES-TEC-FIR	"Final Report"	08/06/23

2.2. Reference Documents

The following table specifies the reference documents that shall be taken into account during project development.

Table 2: Reference documents

Ref.	Code	Title	Issue
[RD 1]	OPENSF-DMS-OSFI-DM	OSFI Developer's Manual	1.22
[RD 2]	PE-ID-ESA-GS-464	ESA Generic E2E Simulator Interface Control Document	1.2.5
[RD 3]	OPENSF-DMS-TEC-SUM01	openSF System User Manual	4.3

2.3. Standards

The following table specifies the standards that shall be complied with during project development.

Table 3: Standards

Ref.	Code	Title	Issue
[SD 1]	ECSS-E-00A	ECSS Space Engineering – Policy and Principles	
[SD 2]	ECSS-M-00-02A	ECSS Space Project Management. Tailoring of Space Standards	
[SD 3]	ECSS-E-10	ECSS Space Engineering: Engineering – Part 1: Requirements and Process	
[SD 4]	ECSS-E-40B, draft	ESA software requirement standard ECSS-E40 tailored to small software projects	
[SD 5]	ECSS-E-ST-40C	Space Engineering Software	

3. PROJECT HISTORY

This chapter presents the main organizational issues of the project, mainly, the schedule, the work breakdown structure and the description of the main work packages.

3.1. Project Objectives

ATSI proposal was prepared answering the ESA ITT AO/1-10736/21/NL/KML, "ATSI: Aerothermodynamics Tools for System Integration".

The joint submission of this proposal from **DEIMOS Engineering and Systems S.L.U. (Spain)**, as prime contractor, and the member of its consortium, **Deutsches Zentrum für Luft und Raumfahrt – DLR (Germany)**, **Heron Engineering (Greece)** and **Empresarios Agrupados Internacional S.A. (Spain)** came from the common desire to provide ESA with the most competitive solution to the stated problem and SW product demand, as well as from the perspective of creating an asset that shall enhance our involvement in the future related activities.

According to Article II, Purpose, Convention of establishment of a European Space Agency, SP-1271(E), 2003 "*ESA's purpose shall be to provide for, and to promote, for exclusively peaceful purposes, cooperation among European States in space research and technology and their space applications, with a view to their being used for scientific purposes and for operational space applications systems.*"

Given the above ESA mandate, most ESA's missions are conceived to learn and increase our knowledge of space and to develop related technologies. This means that each ESA mission is designed to achieve a set of mission goals, in most of the cases defined to push forward the boundaries of the European technology capabilities. Learning from real space missions and experiments is in the essence of the ESA mandate and this objective is very relevant in particular to the domains of the ESA Flight Vehicles Engineering and Aerothermodynamics section.

In this respect, the present activity represents a valuable step ahead to pursue such objective. The specific objectives of the activity are described in [AD 2]:

- The objective of this activity is to prototype a toolset to support the integration of numerical system models, by means of individual modules to interface different numerical models
- Transferring and projecting data from one model to another
- Prepare and run the relevant engineering tools, which include at least the disciplines of CAD, aerodynamics, propulsion, trajectories, structures and thermal.
- The toolset shall be made available under an ESA Software Community Licence

Therefore, the main goal of this activity has been to develop an open-source SW tool to allow ESA performing a set of analysis on the following domains: (1) CAD, (2) propulsion, (3) aerodynamics, thermodynamics, (4) trajectories and (5) structures.

To address this challenging goal, the resulting SW framework provides a generalised interface compatible with TAU, Nastran, ANSYS and ESPSS together with a set of capabilities to manage, process, analyse and report the required information by the user and the tools mentioned above.

3.2. Schedule

Figure 3-1 shows the schedule followed for the ATSI project activities.

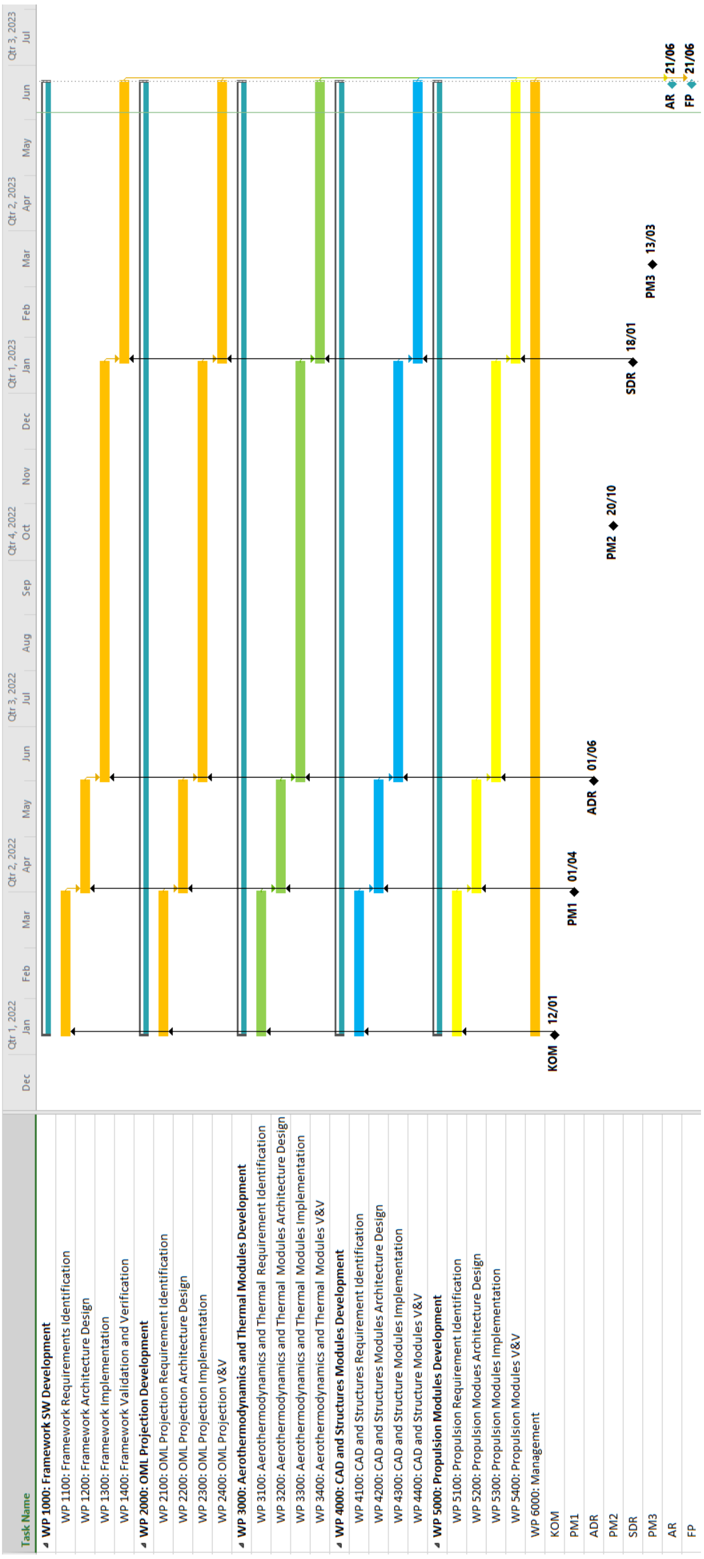


Figure 3-1: ATSI Project Schedule

3.3. Project Team

The following figures show the project industrial consortium:

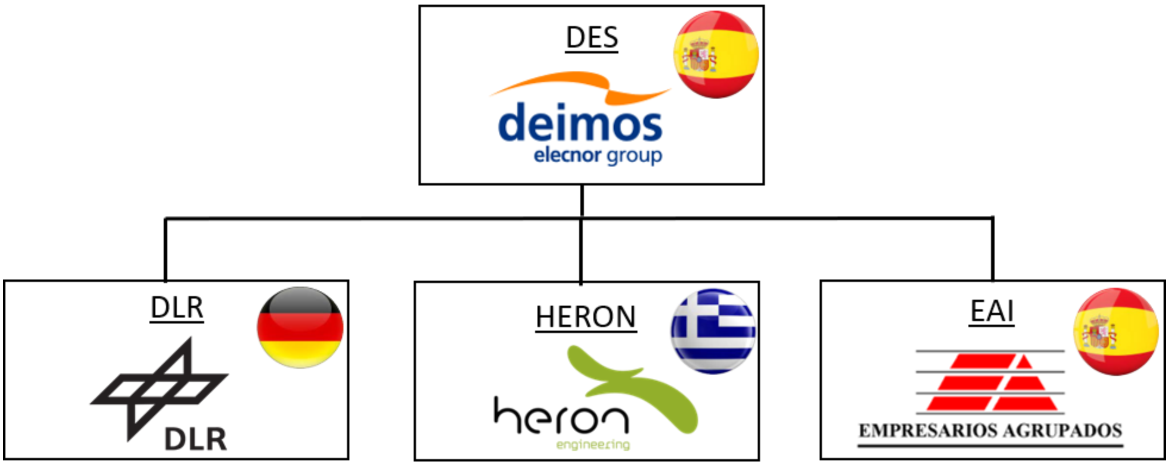


Figure 3-2: ATSI Industrial Consortium

3.4. Work Breakdown Structure

The Work Breakdown Structure was conceived to achieve all the objectives of the study in a timely and effective manner. The followed Work Breakdown Structure is shown in Figure 3-3.

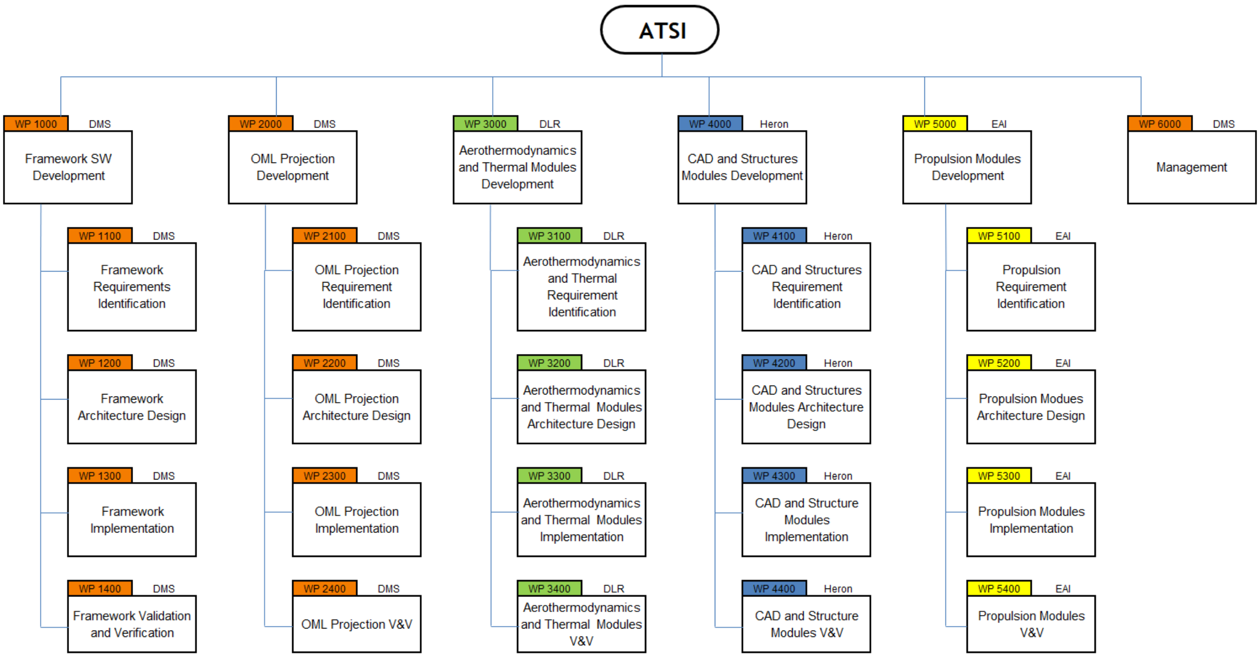


Figure 3-3: ATSI Work Breakdown Structure

3.5. Study Logic

The project study logic plan, which is illustrated in Figure 3-4, has been posed following the premises of the SOW [AD 2], showing the work logic and the tasks of the project with the logical connections among them, the project reviews and the deliverables.

The activity breakdown follows the standard SW development process as per ECSS E-ST-40-C [SD 5].

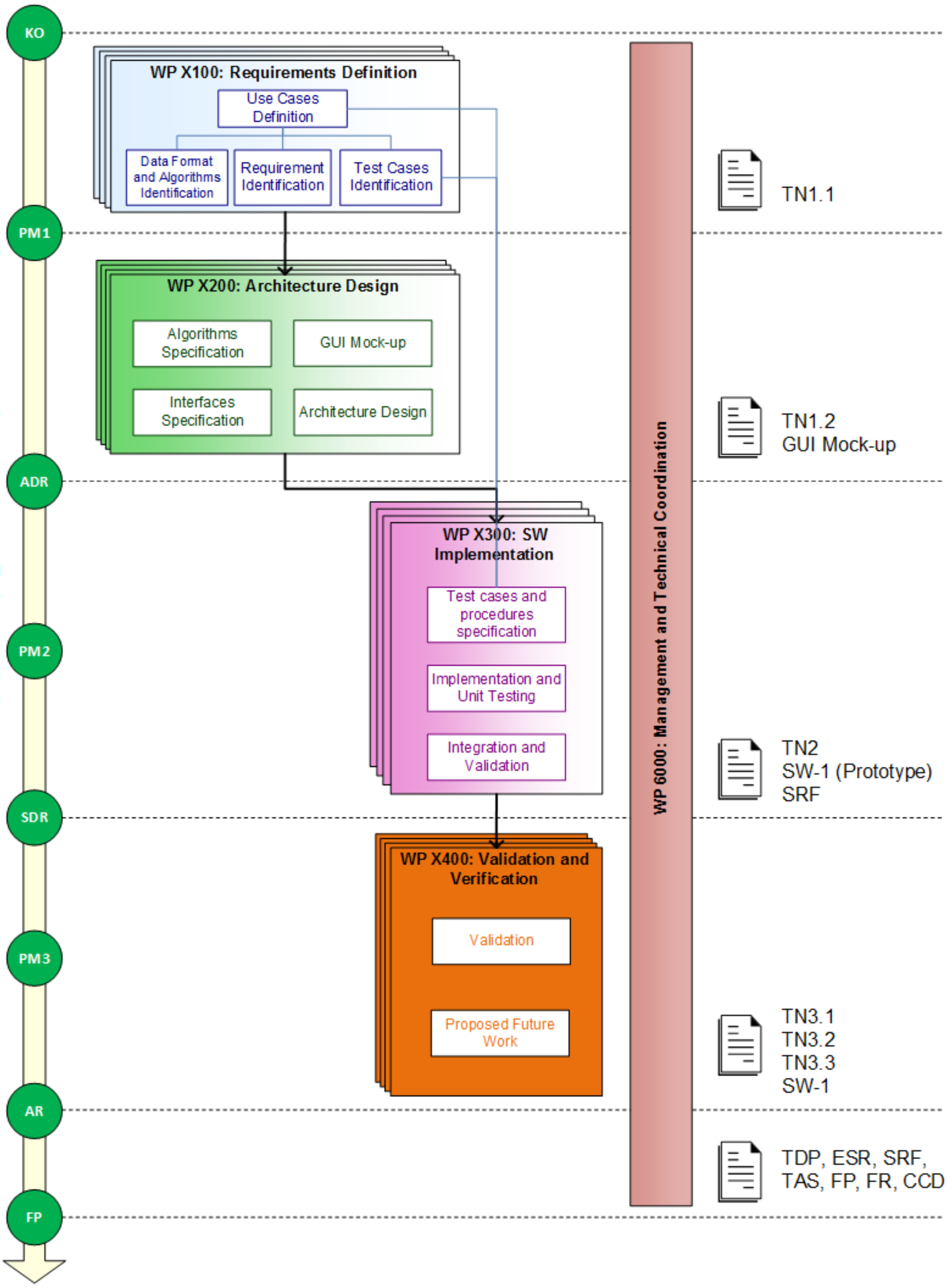


Figure 3-4: ATSI Study Logic

The activities of the project can be seen in two different orthogonal views: grouped according to the standard phases of the SW development process or grouped according to their technical domains. In the WBS (see Sec. 3.4) the latter has been used; however, in the work logic the first approach is preferred, since it better describes the temporal succession of tasks.

According to the WP numbering, the thousands digit is associated to the technical domain (WP 1000 Framework, WP 2000 OML Projection, WP3000 Aerothermodynamics and Thermal, WP 4000 CAD and Structures and WP 5000 Propulsion), while the hundreds digit is associated to the SW development process, as follows:

- WP X100 – Requirement Definition: in these work packages use cases have been defined and specified and from them tool requirements, data formats, algorithms/methodologies and test cases derived.
- WP X200 – Architecture Design: these work packages are meant for deriving a tool architecture design from the requirements specified in WP X100, specifying interfaces, data formats and algorithms/methodologies.
- WP X300 – Software Implementation: during the SW implementation, detailed design has been performed. Test cases and procedures have been specified and the implementation performed, with the support of unit testing.
- WP X400 – Validation and Verification: within this WP verification (testing that the tools calculations are numerically correct) and validation (testing that the tool implements the functionalities it has been designed for, executing the test cases scenarios derived from the use cases) have been performed. The activity has ended with the recompilation of recommendations for future works on ATSI.

Additionally, WP6000 runs in parallel to the whole project lifetime to carry out project management activities to ensure a proper and adequate level of technical and programmatic progress of the tasks. This also covers the prime contractor’s monitoring and control of activities performed by the subcontractor.

At the end of each task a dedicated meeting has been held with the Agency to review the work done and to plan the following activity’s steps.

4. SOFTWARE DEVELOPMENT

This section describes the ATSI development process, giving special attention to the simulator architecture and the validation phase.

4.1. ATSI Life Cycle

From SoW [AD 2], the main objective of this activity *is to prototype a toolset to support the integration of numerical system models, by means of individual modules to interface different numerical models, transferring and projecting data from one model to another, prepare and run the relevant engineering tools, which include at least the disciplines of CAD, aerodynamics, propulsion, trajectories, structures and thermal*.
The tasks specified in the SoW follow the software development waterfall approach, where the corresponding engineering disciplines are arranged sequentially in different phases to facilitate its implementation and control. This is schematically presented in Figure 4-1, where the titles of the boxes represent the process names as defined in ECSS E-ST-40-C, whereas the contents represent the activities as defined in the SoW.

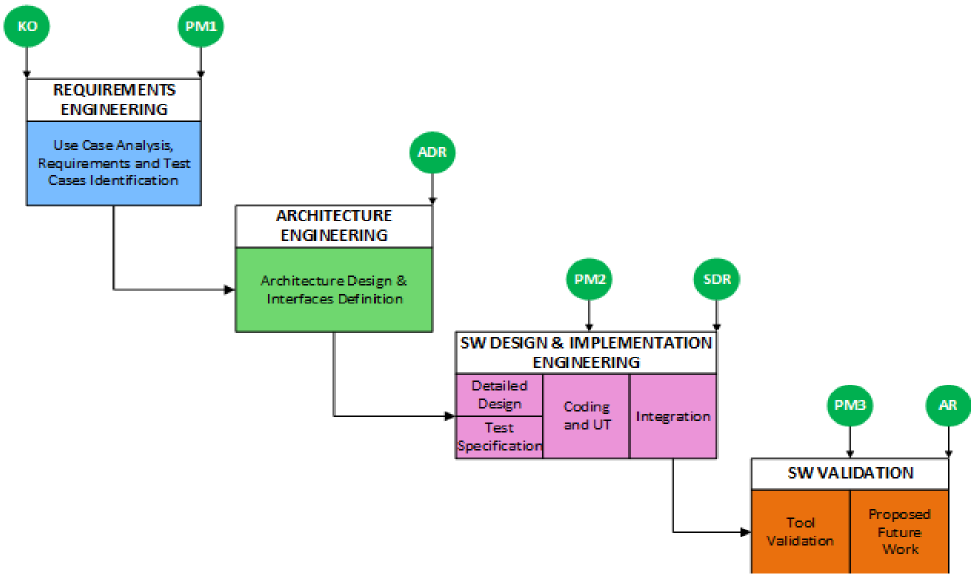


Figure 4-1: ATSI Software Development Life Cycle Process Diagram

4.2. Design and Implementation

The ATSI system is being designed to be a multi-operating system desktop software application with four different functional components:

- The **computational core**, which is composed of six Python modules, related with five functionality groups (readers, writers, grid handling, orchestration and tool launchers) and an auxiliary module with common functionalities.
- The **executable modules**, which are a series of executable scripts that can be interfaced directly with the Graphical User Interface (GUI) and that expose the main functionalities of the six above-mentioned Python modules.
- The **Graphical User Interface**, which provides access to the executable modules and allows the user to build processing chains and manage the configuration and execution of the ATSI analyses. ATSI relies on the ESA openSF integration framework, which was extended and upgraded where necessary to meet the ATSI needs.

- The **Common Data Structure (CDS)**, which can be understood as the glue between the different modules, since it is a container able to store the input and output data of the modules.

This high-level architectural decomposition clearly shows how the tool has been designed for a dual use, either by means of a Command Line Interface (CLI) - as a Python package that provides a set of stand-alone functions - or by means of a GUI.

The resulting high-level system decomposition is depicted in Figure 4-2, where:

- The green dashed line encloses the content of the computational modules. They work with a set of inputs (and configuration files, in yellow) and produce a set of outputs. This represents the CLI use of the tool.
- A module (or a sequence of modules, namely a processing chain) is orchestrated by an integration framework that also provides access to post-processing capabilities to perform automatic report generation accessing the outputs produced by the ATSI modules.

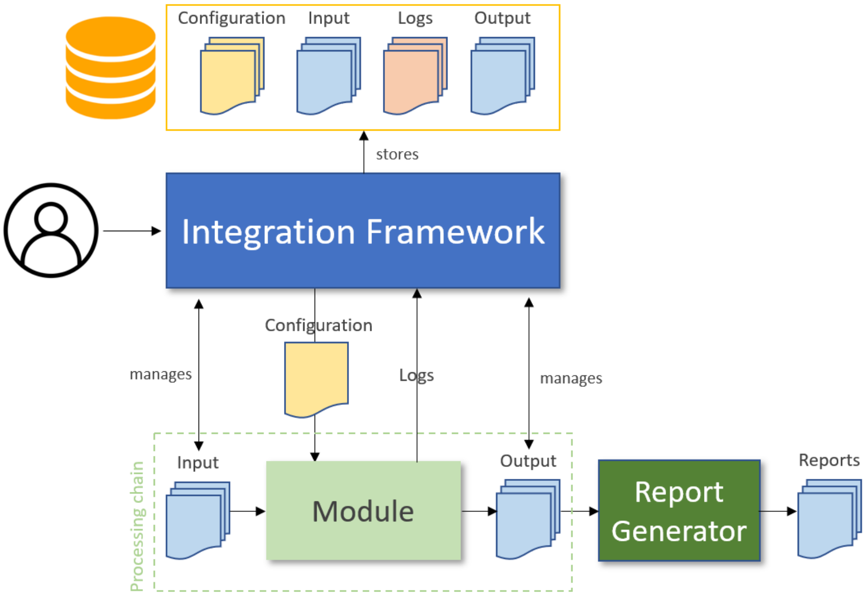


Figure 4-2: PFAT System Decomposition

4.2.1. PFAT re-usability

Due to the high level of synergy between PFAT and ATSI, there are a series of PFAT functionalities and capabilities that are re-used in ATSI, reducing this way the number of duplications between them and increasing the maintainability of the software:

- Both tools are designed for a dual use, either by means of a Command Line Interface (CLI) as a Python package that provides a set of stand-alone functions, or by means of a GUI. Therefore, the design pattern of the PFAT functionalities, strongly based on the openSF simulation framework context manager, was replicated in ATSI.

- The two tools need to communicate large datasets between modules in a common format in order to maximize the flexibility of the user to create processing chains. In PFAT, this goal was achieved by means of the common data structure, which has been used also in ATSI but with the adaptations described in section 4.2.3 below.

- There are PFAT modules that can be directly used in ATSI, such as the TAU reader, the reference trajectory loaders, the trajectory performance, the Tecplot reader, the CDS writer (to write trajectory data to human readable format).

In order to fully benefit from these synergies, some modifications on the PFAT source code have been performed. Among those:

- The architecture of the CDS was re-organized (see section 4.2.3).
- The trajectory performance module in PFAT was modified to provide the body-fixed velocity among the outputs.
- The TAU reader in PFAT was modified to cope with the needs of ATSI.
- PFAT was prepared for its packing as a python wheel. This way, it has been easily distributed and integrated in ATSI, following the same strategy that was used to integrate OSFI both in PFAT and ATSI.

4.2.2. Computational Core

The ATSI computational core is developed in Python and consists of six different modules:

The readers, writers, grid handling, orchestration and tool launchers modules contain the functionalities that implement the ATSI use cases. The common module contains the functionalities that are shared among different modules.

The usage of ATSI as a Python package allows the user to exploit all the capabilities of the tool, including the access to any function in the library and extension or development of new functionalities. It shall be highlighted that the developers can contribute to the ATSI source code using the custom Python syntax for displaying errors, warnings and information messages, which are then directly translated to the GUI thanks to the lightweight interface provided by OSFI. This means that any ATSI function can be used in other contexts and any external function can be used in ATSI.

4.2.3. Common Data Structure

ATSI is capable of handling two types of data, in a broad sense:

- Tool-specific files in their native format
- Common Data Structure (CDS) instances

The CDS is a solution already adopted in the PFAT project with a main difference between the CDS functionalities required by PFAT and ATSI:

- PFAT required to set, store and access data series associated to the evolution of an independent variable (typically time, but not necessarily so).
- ATSI requires to set, store and access data sets with no associated evolution (in time or along any other independent variable). In addition, within ATSI, it is desirable to be able to use the CDS to store only metadata (see for example **UC-100 Trajectory data to metadata**) or only the mesh information (point coordinates and elements connectivity, see for example **UC-020 Load CAD data**) without the need to add any other data.

Therefore, and in order to exploit this synergy, an architectural change of the PFAT CDS has been performed within the ATSI project, consisting on having different types of common data structures depending on the data stored: one type to store only metadata, another one to store static grid data (the one required in ATSI) and another one storing data series that can be associated or not to a grid (the one in PFAT).

4.2.4. Executable Modules

The executable modules in ATSI represent standalone executable applications that make use of the computational modules, providing them with the possibility of being run directly as python executables and of being integrated in the GUI either as they are or within a computational chain, the latter representing a complex analysis.

Both the executable modules and the GUI implement an error handling system in charge of intercepting the error and log messages, and to manage them, either presenting them to the user or taking the necessary actions. The communication between the executable modules and the integration framework was simplified by using the openSF Integration Library (OSFI), a collection of functionalities that helps with the integration of the modules within openSF. The complete map of executable modules included in ATSI is shown in Figure 4-3.

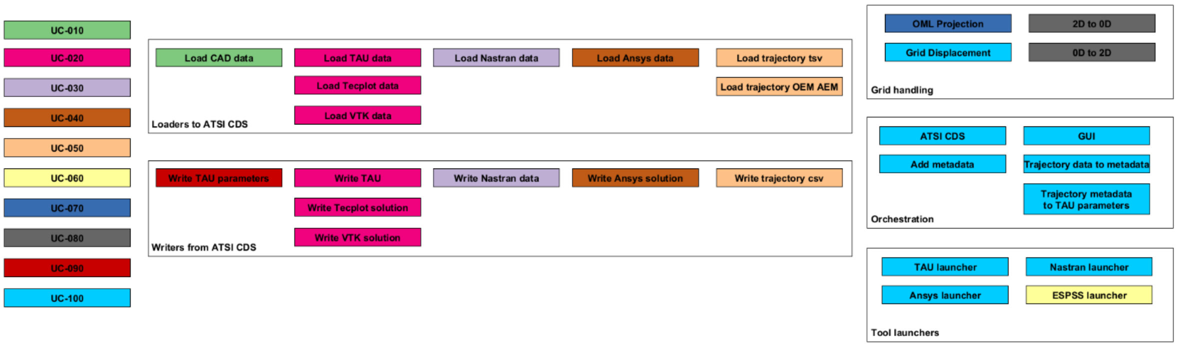


Figure 4-3: ATSI functionalities map

4.2.5. Graphical User Interface

ATSI makes use of an ESA open-source simulation framework developed using Eclipse RCP, called openSF, as framework for the orchestration of the processing chains.

openSF is a software framework aimed at supporting a standardised end-to-end simulation capability allowing the assessment of the science and engineering goals with respect to the mission requirements. Scientific models and product exploitation tools can be plugged in the system platform with ease using a well-defined integration process.

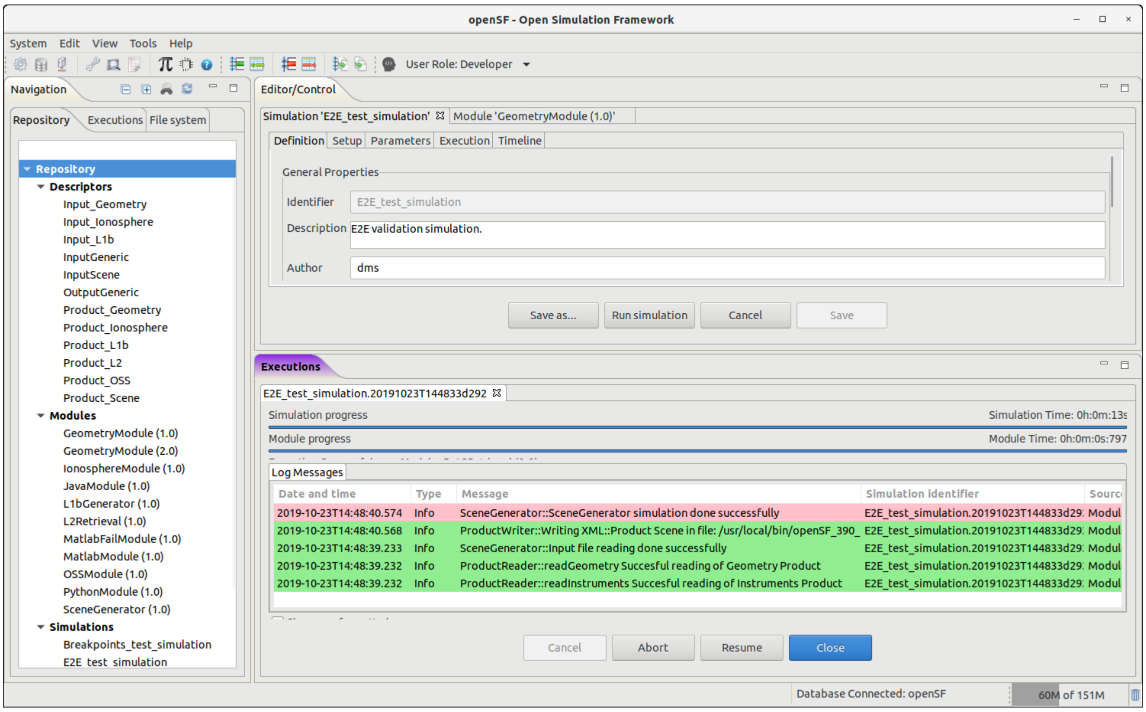


Figure 4-4: openSF appearance

Some of the advantages provided by the use of openSF can be summarised as follows:

- ❑ openSF already covers some of the requirements identified for ATSI, such as the capability to define and orchestrate workflows with clear identification of computational modules and interfaces among them. This allowed to focus the development activities towards all the ATSI specific feature, re-using all the generic functionalities already provided by openSF.
- ❑ openSF is distributed under the ESA Software Community License - Type 3, an open-source licensing scheme compatible with the ATSI requirement to be open source.

- openSF provides a reliable framework, already under development for more than 10 years and with updates and bug fixes publicly made available to the community approximately every six months. This is a great asset, since it would provide the project with an already mature framework, with many early-life glitches and bugs fixed.
- openSF is developed, maintained and distributed to the same target operating systems for which ATSI is being developed.

The reference baseline version of openSF considered for ATSI is openSF-4.1 but some specific improvements have been included as additional features for ATSI. The main adaptations are the support for circular chains and the automatic adaptation of databases created with previous versions of openSF.

4.2.5.1. Support for circular chains

A new option called "Run sequentially" has been added in the "Execution" tab of the simulation setup in openSF. This option, if selected in combination with the iteration mode, allows forcing each iteration to wait for the previous one to be complete before starting, thus creating a circular chain.

4.2.5.2. Automatic upgrade of database backups

A new feature of the framework allows to automatically convert old versions of the openSF databases (created with openSF versions previous to 4.0) to the new database format. This feature is specifically useful since PFAT used an old version of the database format, so this modification of the framework allows the compatibility with the processing chains defined in PFAT also with the ATSI GUI. The upgrade feature is automatically enabled with no user intervention required.

4.3. Validation and Verification Approach

In order to guarantee a robust toolkit, complying with both software specifications and user requirements, ATSI has been tested through an extensive validation campaign.

During the implementation phase, all the functionality developed were tested with unit testing that covered 92% of the code by means of 167 unit-tests. Such a large testing guaranteed an elevated reliability of the implemented functions that minimised risks at integration level and paved the way for the validation and verification campaign.

The validation and verification campaign has been divided in two different parts:

- The integration tests
- The acceptance tests

Integration tests

Such tests have been implemented before the closure of the implementation phase and they targeted the early detection of integration issues between different modules in the toolkit. They did not focus on the numerical results of the different modules but rather on the possibility to properly exchange data between two (or more) different modules.

Acceptance tests

The acceptance tests have been defined through some real use case scenarios that ATSI could be used for. They are:

- ATS-01: Chaining CFD analysis with propulsion system analysis with ESPSS
- ATS-02: Chaining CFD analysis with thermo-structural analysis. This acceptance test was finally split into two similar versions, one using Nastran for thermo-structural analysis and the other one using Ansys.
- ATS-03: Loading a grid from CAD file and projecting CFD results on it.

The acceptance tests use realistic flight data obtained through high fidelity simulators.

The whole validation campaign has been conducted on the two target platforms, namely: Windows 10 and Linux Ubuntu 20.04 LTS.

5. ATSI CAPABILITIES

The current section provides a high-level description of the data flows identified within the system and the overall capabilities provided by the toolkit.

These data flows constitute the baseline of the processing chains defined for the acceptance tests. These interactions are illustrated in terms of the different scientific domains: trajectory, grid handling, aerothermodynamics, thermo-structural, propulsion.

In the following subsections, the data-flows will be described with diagrams that use the following colour convention:

- ❑ Fuchsia: file in non-standard ATSI format, generated from/to an external tool or flight data
- ❑ Green: ATSI processing module
- ❑ Lilac: file in standard ATSI format (CDS)

5.1. Trajectory (UC-050 and UC-100)

The modules focusing on the trajectory domain in ATSI belong to use cases 050 and 100.

As shown in Figure 5-1, the output of the trajectory loading modules inside UC-050 is a CDS with the trajectory data and it can be used as input for the UC-100 modules or for any method of the CDS API existing in PFAT.

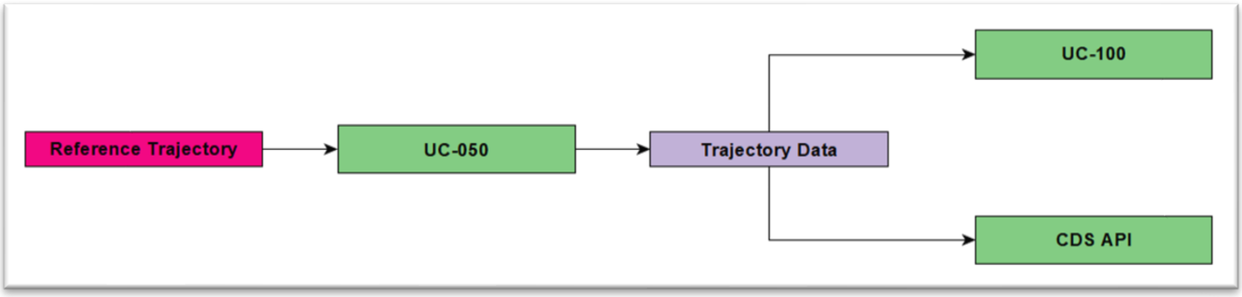


Figure 5-1: UC-050 and UC-100 interactions

As shown in Figure 5-2, the output of the trajectory part of UC-100 can be used as an input for UC-090.

Figure 5-2: UC-100 interaction with UC-090

5.2. Grid handling (UC-010, UC-070, UC-100)

The modules focusing on the grid handling in ATSI belong to use cases 020, 040 and 090.

Some of the interactions between these modules and with other modules from other domains are here illustrated. Figure 5-3 shows how UC-010 and UC-070 interact with each other: the output of UC-010 provides the destination grid for the projection, and the data can be provided by any CDS with grid data associated.

Figure 5-4 shows the interaction between UC-070 and the modules from the aerothermodynamics and thermo-structural domain. The results of a Nastran analysis, loaded with the Nastran loader from UC-030, are projected onto the grid defined by TAU (read by UC-020). The output of this projection can then be used as input for the TAU writer (UC-020). Similar interaction can be applied using ANSYS instead of NASTRAN.

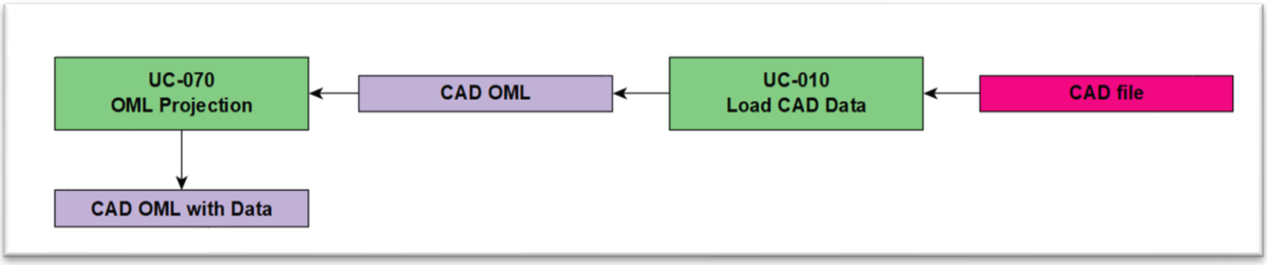


Figure 5-3: UC-010 and UC-070 interaction

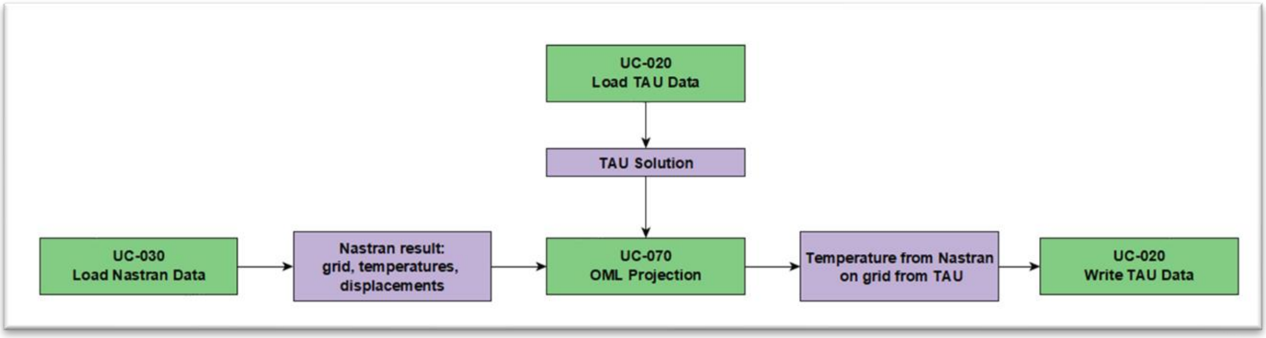


Figure 5-4: UC-070 interaction with UC-020 and UC-030

Finally, some interactions of the grid displacement module in UC-100 are illustrated. As shown in Figure 5-5, the grid displacement module can operate both on thermo-structural results that contain a displacement field (which is used to move the original grid) or on any other data set. In case this dataset does not contain a displacement field, the user can specify the displacement values to be applied.

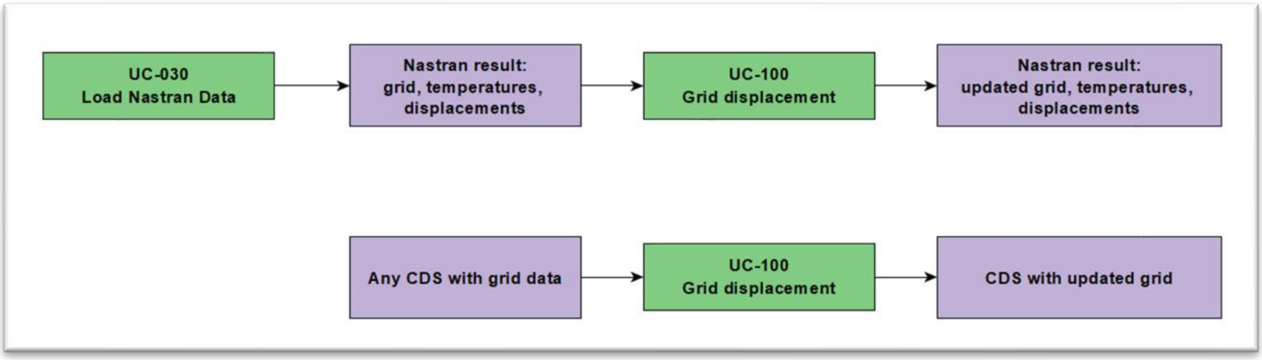


Figure 5-5: Possible interactions of UC-100 Grid displacement

5.3. Aerothermodynamics (UC-020, UC-090 and UC-100)

The modules focusing on the aerothermodynamics domain in ATSI belong to use cases 020, 090 and 100.

The interactions of UC-020 have already been illustrated in the previous section in terms of TAU reader and writer. Here, the interactions of UC-020 in terms of VTK and Tecplot data writing are shown. The modules are designed in such a way that any CDS with data and grid can be used as input for both modules (Figure 5-6), meaning that UC-020 can interact with most modules in the ATSI tool.

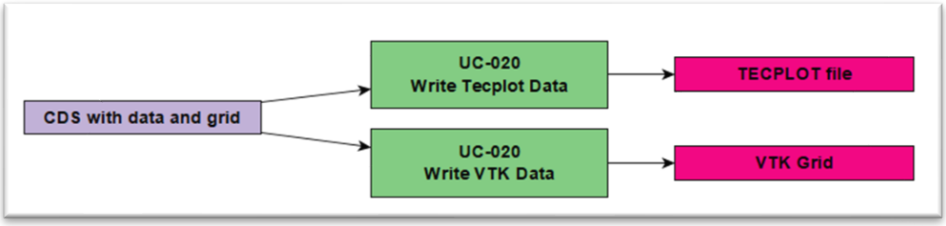


Figure 5-6: Interactions of UC-020 in terms of Tecplot and VTK data writing

A possible interaction between UC-100 and UC-090 has already been illustrated in the trajectory section. Here, a more detailed illustration of the possible interactions of UC-090 is presented.

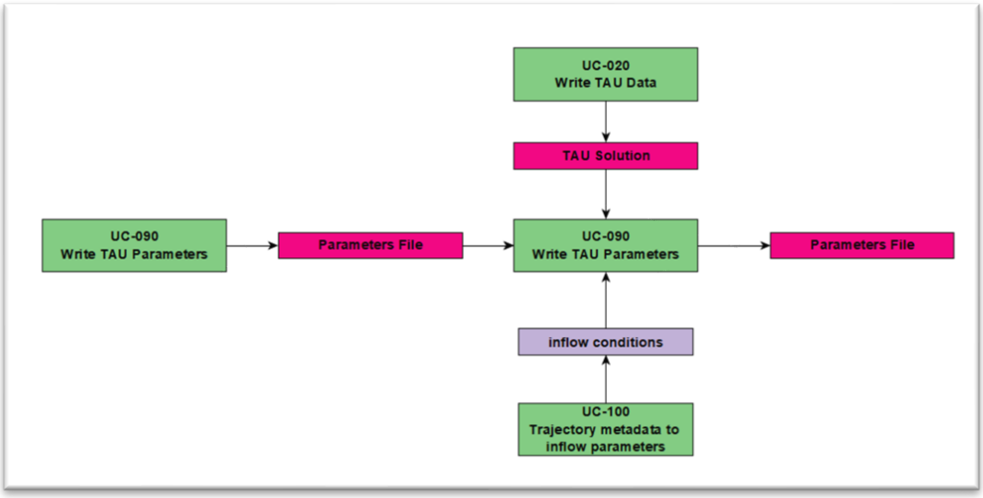


Figure 5-7: Interactions of UC-090

Finally, Figure 5-8 shows another possible interaction between UC-090 and UC-100 which is offered by the fact that UC-090 prepares the TAU parameters file used to execute the CFD analysis with TAU. Such analysis produces outputs which are then read with UC-020.

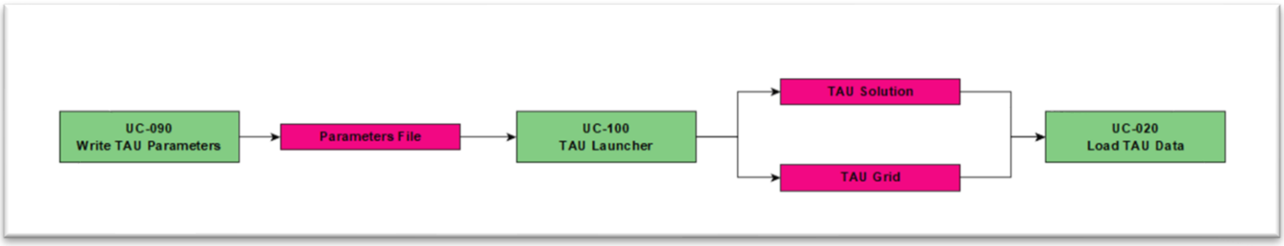


Figure 5-8: Interactions concerning TAU analysis execution

5.4. Thermo-structural (UC-030, UC-040 and UC-100)

The modules focusing on the thermo-structural domain in ATSI belong to use cases 030, 040 and 100.

Some of the interactions of these modules have already been shown in Figure 5-4, concerning the OML projection. Here, the interactions with UC-100 in terms of launchers are presented.

Figure 5-9 shows the interaction between UC-030 and UC-100. The Nastran data writer is used to write the input .bdf files for the Nastran execution, which is then launched with the Nastran launcher in UC-100. The outputs of the simulation are then loaded into CDS with the Nastran data loader in UC-030.

Figure 5-10 illustrates the interaction between UC-040 and UC-100. The Ansys data writer is used to write the input file for the Ansys execution, which is then launched with the Ansys launcher in UC-100. The outputs of the simulation are then loaded into CDS with the Ansys data loader in UC-040.

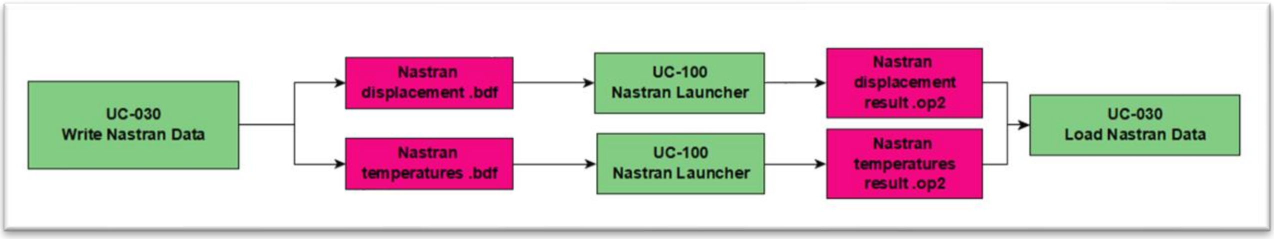


Figure 5-9: UC-030 and UC-100 interaction

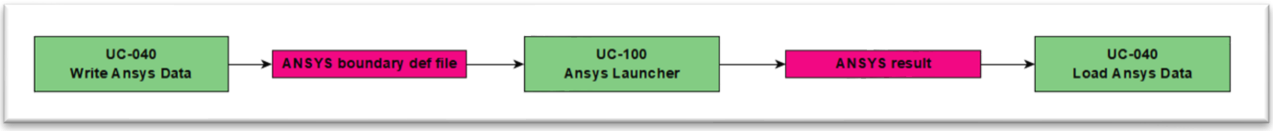


Figure 5-10: UC-040 and UC-100 interaction

5.5. Propulsion (UC-060, UC-080)

The modules focusing on the propulsion domain in ATSI belong to use cases 060 and 080.

Figure 5-11 shows the interaction between UC-080 and UC-060 and their interaction with UC-020. The output of a CFD analysis (loaded with UC-020) can be used as input for the 2D-to-0D module to retrieve the averaged inlet data. This is then used as an input for the ESPSS launcher in UC-060 and the output of this is then projected to the required output grid by the 0D-to-2D module. In this diagram, the output grid is provided by a TAU solution (loaded with UC-020) but it could be any other CDS with grid data. For instance, it could be the output of UC-010 (Grid data loaded from an .STL file).

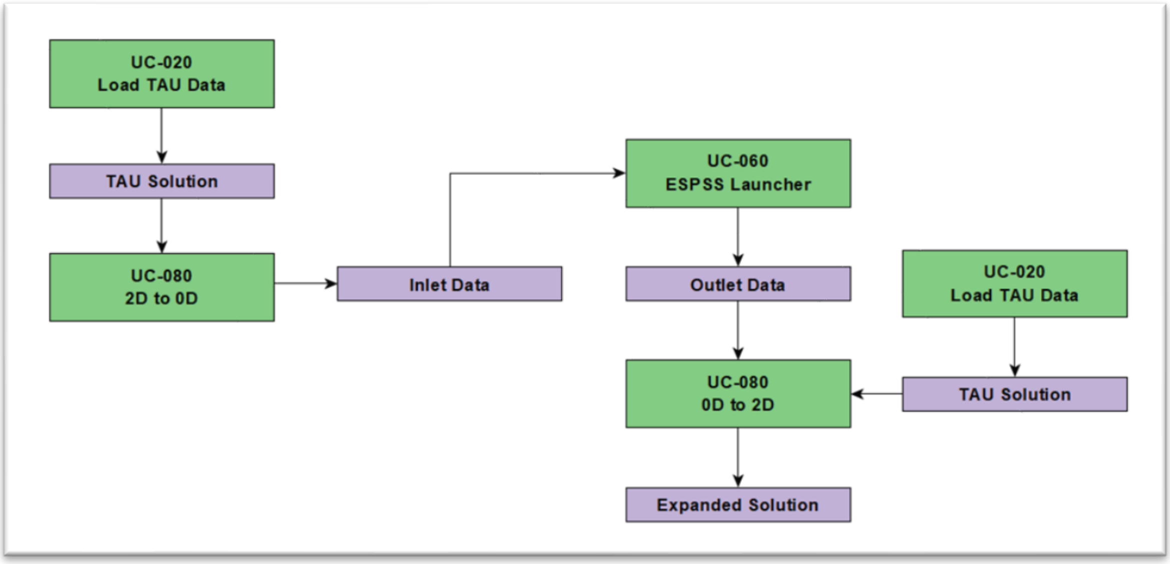


Figure 5-11: Interaction between the propulsion modules

6. ATSI VALIDATION CAMPAIGN

As previously explained, the ATSI validation campaign includes three acceptance tests, which are illustrated in the following sections.

6.1. ATS-01

6.1.1. Test specification

The purpose of ATS-01 is the validation of the mapping functionalities 2D to 0D and 0D to 2D and the execution of the ESPSS model.

Reference trajectory data for this test is based on a representative re-entry scenario. DEIMOS has a strong background in re-entry analysis for several vehicle's classes (e.g., capsules, lifting bodies, reusable launchers) and therefore has different in-house high-fidelity simulators and data available to synthetically generate representative data for re-entry scenarios including relevant and required atmospheric variables such as pressure, density, temperature. The final selection of the scenario has been specified and detailed in [AD 3].

The grid data for this test is based on an air-breathing engine, similar to the SABRE Engine, represented in Figure 6-1, that is representative of the engine model identified in the ESPSS software. The engine model used for the analysis has been modelled as a half-model, exploiting its symmetry properties, in order to reduce the computational effort for the CFD analysis.

The scenario to be tested is characterised by the following parameters:

- Mach number: 5.785
- Density: 0.4133 kg/m^3
- Temperature: 223.25 ºC

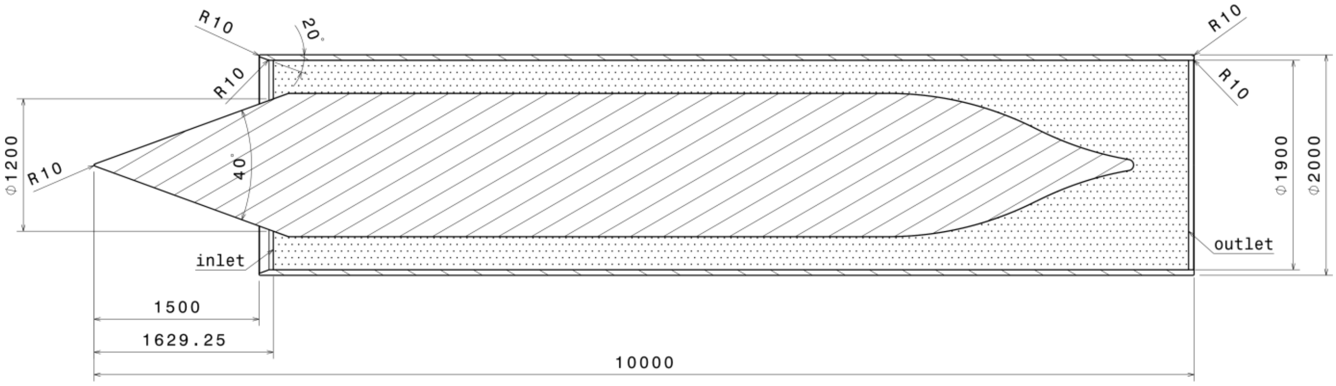


Figure 6-1: ATSI test Engine

6.1.2. Test results

The trajectory has been retrieved from a high-fidelity simulation tool available at DEIMOS, typically used for re-entry analyses. It has been cut in time because the test only focuses on a single time step. The test can be executed both via command line interface and from the ATSI GUI.

The successful execution of ATS-01 concludes with the successful projection of the zero-dimensional flow properties at the engine outlet onto the 2D grid representing the engine exit surface.

Considering the engine axis as the X axis, Figure 6-2shows the velocity field in the YZ plane. The inlet of the engine is ring-shaped and the velocity distribution can be observed to be in line with the expected field for this type of grid. The flow velocity is null at the walls and it increases rapidly towards the central part of the ring. The velocity value increases very sharply close to the walls, thus the null values are not visible in the picture but the overall distribution is physically correct.

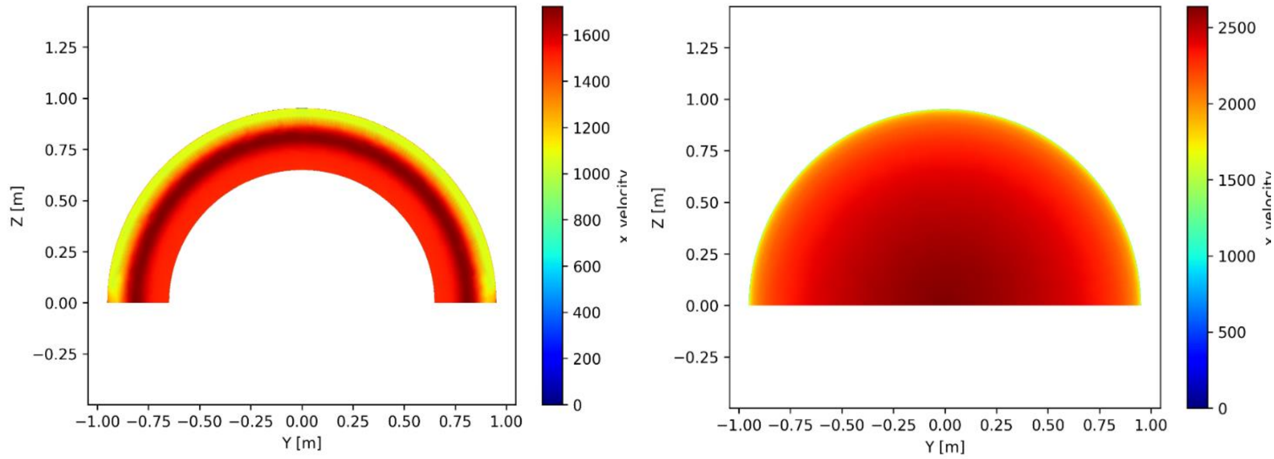


Figure 6-2: ATS-01 flowfields at engine inlet and outlet

The final step of the chain is the projection of the engine outlet properties onto the grid representing the engine outlet surface. This step is performed with the 0D-2D projection module and the output can be seen in Figure 6-2. The distribution, obtained with the power law for the velocity field, has its max value at the engine axis and the minimum value (eventually, zero) at the external wall.

6.2. ATS-02

The purpose of ATS_02 is the implementation of the workflow described in UC-100.

As per the previous test, reference trajectory data for this test are based on a representative re-entry scenario. DEIMOS has a strong background in re-entry analysis for several vehicle's classes (e.g., capsules, lifting bodies, reusable launchers) and therefore has different in-house high-fidelity simulators and data available to synthetically generate representative data for re-entry scenarios including relevant and required atmospheric variables such as pressure, density or temperature. The final selection of the scenario has been specified and detailed in [AD 3].

The grid and simulation data for this test is based on the ENTRAIN model, represented in Figure 6-3.

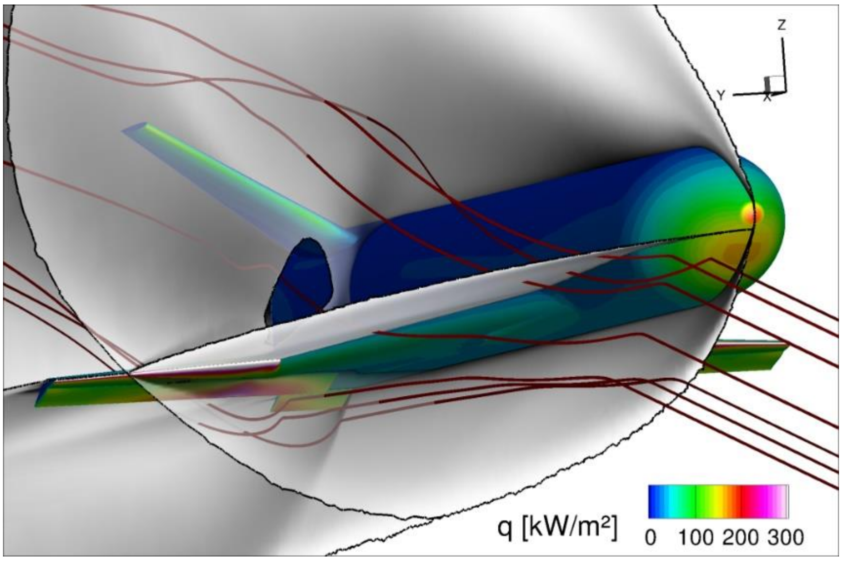


Figure 6-3: ATS-02 Flowfield of DLR ENTRAIN

In order to improve the validation, this test is split into two different versions that perform similar steps but using different tools:

- ATS-02a uses TAU for the aerothermodynamics analysis and NASTRAN for the thermal/structural analysis.
- ATS-02b uses TAU for the aerothermodynamics analysis and ANSYS for the thermal/structural analysis.

6.2.1. ATS-02a results

The trajectory has been retrieved from a high-fidelity simulation tool available at DEIMOS, typically used for re-entry analyses. It has been cut in time because the test only focuses on two timesteps. The test can be executed both via command line interface and from the ATSI GUI.

The successful execution of ATS-02a concludes with the successful execution of the second loop iteration. This implicitly validates the capability of ATSI to work with closed-loop simulations (circular chains, like in a FSI simulation).

Figure 6-4 shows the TAU grid with the associated heat-flux distribution. It is evident how the mesh is much finer closer to the leading edge of the wing to allow modelling the flow field distribution accurately, and consequently the heat flux distribution.

The heat-flux field (together with the pressures field) is used to feed a Nastran simulation. The results of this simulation are the temperature and the displacement fields generated by the heat-flux and pressure loads. In order to feed this back to TAU, this temperatures field has to be projected back to the fine CFD grid. The outcome of this projection can be seen in Figure 6-5.

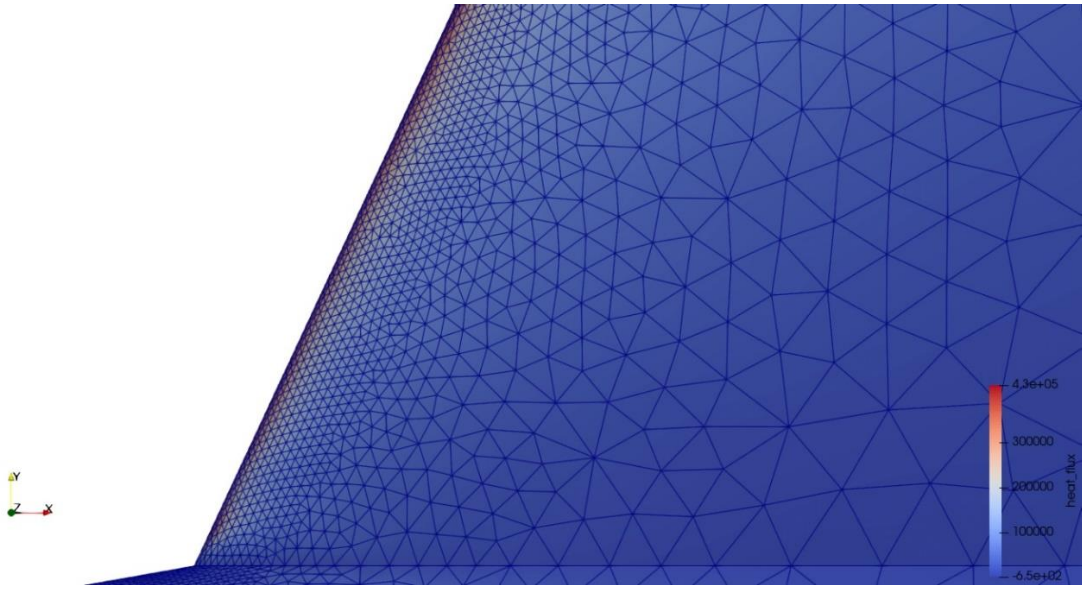


Figure 6-4: Detail of the heat-flux distribution after the first TAU analysis

Patterns are fully in line with the expected performance of the chosen projection algorithm. Indeed, this test case is an extreme case due to the very large difference in elements size between the two meshes. Under such conditions, the performance of the algorithm shows a high level of robustness.

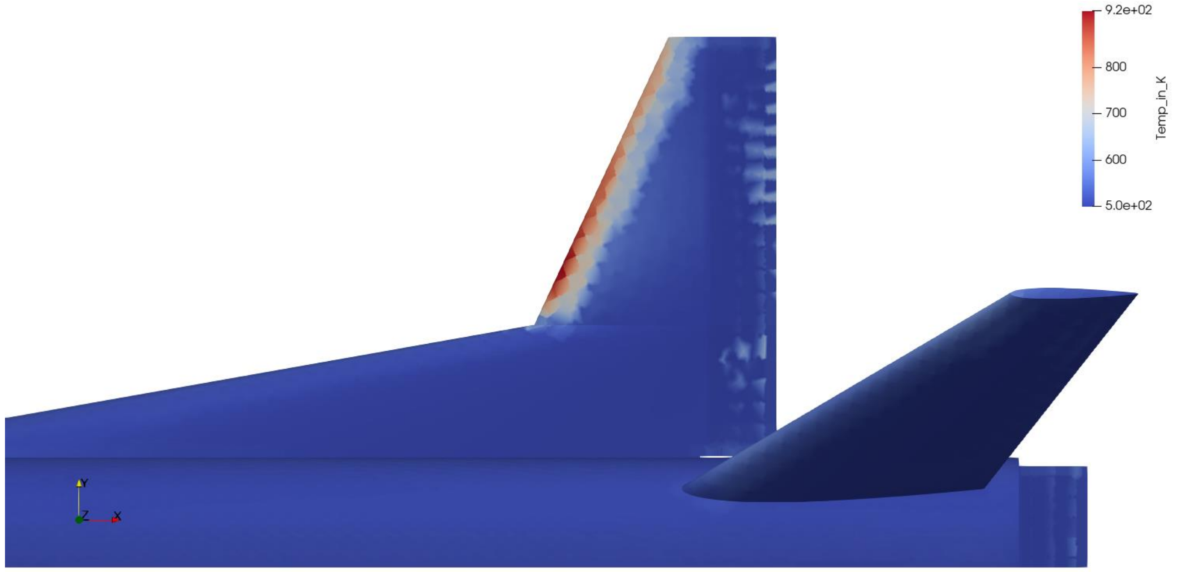


Figure 6-5: Nastran solution for the first iteration projected to the CFD grid, temperature field

6.2.2. ATS-02b Results

ATS-02b aims at providing a term of comparison for the results of ATS-02a. To make the results comparable, the initial conditions for the two chains are exactly the same. The Ansys grid has been created identical to the Nastran grid to additionally improve the comparability of the two analyses. The first difference between the two chains lies therefore in the Ansys analysis. After that, the Ansys results are projected back to the CFD mesh and the outcome of this projection is shown in Figure 6-6. It is clear how the same pattern observed in ATS-02a can be seen here.

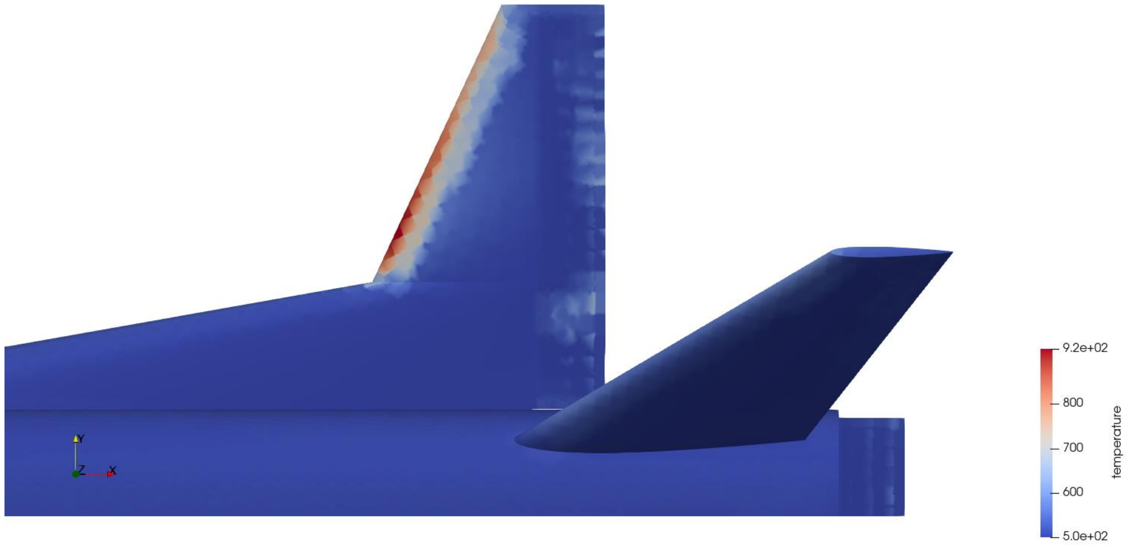


Figure 6-6: Ansys solution projected onto the CFD grid, temperature field

The comparison of the results of the first loop iteration, allows to draw important conclusions concerning ATSI validation. Thanks to the adoption of common way of storing and handling datasets (the Common Data Structure), the outputs of the CFD analysis can be used for any of the two structural/thermal tools

(Ansys or Nastran), without requiring a specific adaptation. In other words, the Nastran writer and Nastran launcher can be substituted in the chain by the Ansys writer and the Ansys launcher and the chain keeps working properly (provided that the configurations are adapted). In the same way, the Nastran results (read with the Nastran reader) can be substituted by the Ansys results (loaded with the Ansys reader) and both outputs can be used to feed-back another iteration of CFD simulation.

This is a key aspect in the validation of ATSI, since one of the main goals of the tool is the development of a modular architecture in which the way data are stored and transmitted and the way the chains are defined does not strictly depend on the modules that are used.

6.3. ATS-03

6.3.1. Test specification

The purpose of ATS-03 is to use an OML generated from CAD data as destination OML in the OML projection algorithm. The CAD data used for this test, next to the TAU and grid data, are based in the same model from ATS-02, which are already available to the consortium.

6.3.2. Test results

Figure 6-7 shows the comparison between the heat flux field at the nose on the CFD grid, which is the original grid associated to the heat flux data, and the same field projected onto the CAD grid loaded from file.

It is clear that the heat flux distribution is very accurate and the projection does not include any distortion with respect to the original dataset. This is due to the fact that the two meshes are comparable, highlighting the importance of producing structural/thermal meshes that are capable of coping with the very sharp distributions of the flow properties of a CFD analysis close to the most critical portions of the surface (e.g., the leading edges).

a

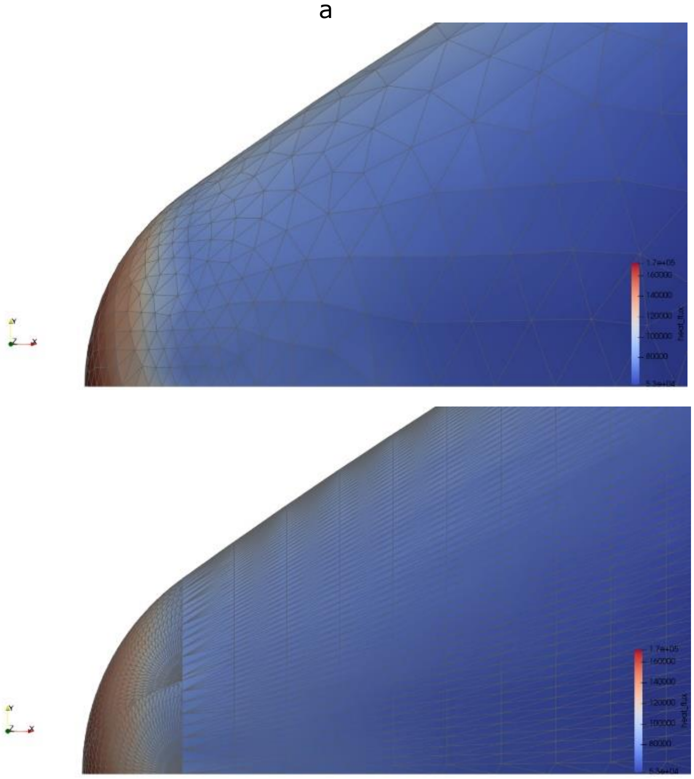


Figure 6-7: Heat flux at the nose on CFD and CAD grids

7. CONCLUSIONS AND FUTURE PESPECTIVES

7.1. Achievements of ATSI Project

During the ATSI activity, the following specific objectives have been fully achieved:

- A dedicated toolkit for the integration of different aerothermodynamic tools has been designed and implemented, allowing the user to define a set of modules and the connection between them to perform sequential analyses given by a chain. This chain, defined by the user, can either be an open chain or even a circular chain, allowing ATSI to define a loop (usually in time) to repeat the analyses at every point a trajectory for instance, using as input for the next step, the calculated outputs from the previous one.
- A series of robust algorithms have been implemented, allowing ATSI to load and write different set of files from very different tools and domains, such as CAD files, trajectory files, structural, thermal, CFD…
- The execution and interoperability of the tools has been guaranteed by the definition of a set of algorithms oriented to load and execute different engineering software tools commonly used by ESA (e.g., TAU, NASTRAN, ANSYS, ESPSS);
- In order to assure the execution of different tools in a sequential way, projection algorithms have been designed and implemented, allowing ATSI to map the results coming from a particular tool to the CAD/input files of the next ones, assuring that the information is coherent among the analyses run in a chain.
- A standard exchange format coming from PFAT (the Common Data Structure) has been improved and implemented, to use the toolkit interoperability.
- A powerful Graphical User Interface is provided along with a flexible Command Line Interface, providing a dual approach to ATSI use;
- All the points above allow ATSI to orchestrate a workflow defined by the user, in which different independent modules can be selected to generate a chain and provide different set of results from different disciplines, always assuring that the information is correctly transmitted among them thanks to the Common Data Structure.

7.2. Future Work

ATSI is a **proof-of-concept** project which has been designed in the first place and then implemented with the goal of demonstrating the feasibility of chaining large sequences of simulations from different external tools which are typically executed separately both due to the different interfaces and different grids adopted. It has proven effective in achieving this goal as full end-to-end analyses have been performed with different external tools in the loop (TAU + Nastran, TAU + Ansys, TAU + ESPSS).

Despite this successful demonstration, this first implementation of the tool paves the way to potential future developments both in the direction of improving the user ease-of-use and increasing the technological content. Details on this possible future work have been explained in [AD 7].

End of Document