Acta Astronautica 214 (2024) 253–260

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Open-source MBSE workflow for automated spacecraft thermal analyses from a system model IAC-22,D1,4B,2,x70017

Henning Heibrok [a,*,1], Anton Donner [a], Moritz Edelhäuser [a], Tobias Franz [b]

[a] Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) – Galileo Competence Center, Münchener Str. 20, Weßling-Oberpfaffenhofen, 82234, Germany
[b] Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) – Institute for Software Technology, Lilienthalplatz 7, Braunschweig, 38108, Germany

A R T I C L E I N F O

Keywords:
Model-based systems engineering (MBSE)
Thermal analysis
Simulation

A B S T R A C T

Model-based systems engineering (MBSE) is an emerging methodology not only in the field of spacecraft systems engineering. The key objective is to store the whole system design in one single source of truth system model. The continuously maintained and customizable open-source MBSE tool Virtual Satellite developed by German Aerospace Center (DLR) offers a framework for storing such a system model. Since the open-source 3D computer aided design (CAD) software FreeCAD has a dedicated Virtual Satellite workbench for importing the 3D model file generated by Virtual Satellite, it is possible to export the 3D model from Virtual Satellite, apply changes to it in FreeCAD, and afterwards reimport it in Virtual Satellite. This paper presents a workflow implementation to conduct automated steady-state and transient thermal analyses using the Virtual Satellite system model as a basis, utilizing the aforementioned features. Part of this effort was to extend Virtual Satellite by thermal modeling capabilities, such as material parameters, surface characteristics, thermal interfaces, boundary conditions and further relevant aspects. With custom Java applications, so-called "Apps", all aspects of the system model relevant for a thermal analysis are exported from Virtual Satellite and written to accordingly generated text files. Moreover, the before described 3D model export is used to export the model to FreeCAD. The FreeCAD Python console, in combination with the finite element method (FEM) workbench, is used to create a script that executes meshing, application of the boundary and initial conditions, as well as identification of contacts between parts and application of the specified contact conductance values. The obtained files form a complete thermal model and can be fed directly into the open-source FEM software CalculiX for executing the thermal analysis. Then, the result can be processed by an additional Virtual Satellite App. To account for the space thermal environment an additional feature was added to the FreeCAD script. The feature uses orbit information obtained from external mission analysis tools to calculate the approximate combined heat load from Sun, Earth infrared, and albedo radiation at each time point in the specified simulation time interval. This new workflow allows to execute quick thermal analyses at an early design stage and therefore opens new opportunities in the evaluation of different system designs, as early as in a concurrent engineering study. Configuration parameters are provided that also allow more sophisticated analyses in later development phases, such as custom mesh definition and local mesh refinement.

1. Introduction

Modeling a system allows to manage its complexity better by giving different viewpoints, utilizing abstraction and linking of dependencies and constraints. Facilitating model-based systems engineering (MBSE) additionally enables rapid, low-effort system analyses and is particularly useful in the early design and development phase of a space project. Inputs from numerous engineering disciplines influencing each other have to be combined and aligned in the system design. An MBSE single source of truth system model helps by ensuring that all involved disciplines share the same up-to-date model of the system at any time, preventing errors being introduced from engineering processes using outdated data. Especially in concurrent engineering studies, which are very common for space projects during the earliest development phases, this is crucial, as it supports an agile like development where the system design is often iterated in order to find feasible designs. Making the latest design available to all stakeholders in a central repository

* Corresponding author.
E-mail addresses: Henning.Heibrok@dlr.de (H. Heibrok), Anton.Donner@dlr.de (A. Donner), Moritz.Edelhaeuser@dlr.de (M. Edelhäuser), Tobias.Franz@dlr.de (T. Franz).

[1] Major parts of this work were conducted as Master's Thesis at the University of Bremen.

https://doi.org/10.1016/j.actaastro.2023.10.024
Received 8 March 2023; Received in revised form 31 July 2023; Accepted 14 October 2023
Available online 24 October 2023

improves the efficiency of these iterations. With configuration and version control management, engineers can still decide at which time the changes are updated in the central system model and a rollback to past versions is always possible. Concurrent engineering studies and the corresponding data exchange is further described in [1,2].

One of the involved disciplines is thermal engineering. Its objective is to find suitable solutions ensuring that temperatures throughout the spacecraft stay within the ranges of the respective units' and components' operating temperatures. An inadequate thermal design may lead to overheating or under-cooling and therefore to failures and potentially to loss of functionality. Therefore, it is important to confirm the feasibility of a thermal design concept as early as possible. This prevents not only potential failures, but can also avoid redesigns at later stages, which may consume a lot of resources in terms of time and cost.

Since in-depth thermal analyses with simulations are cumbersome and therefore not often utilized in the fast changing environment of the earliest project phase, thermal engineers are often restricted to parametric approximations for finding an initial thermal design concept. The first computational models are usually created in phase B, when the overall design is already advanced to some extend [3]. Phase B is the third mission phase according to European Cooperation for Space Standardization (ECSS). It covers the preliminary design of the space craft and is following the phases 0 and phase A. Phase 0 covers mission analysis and need identification; phase A, feasibility evaluation and initial system concept and technical solution definition [4]. In many cases the effort required for manually setting up a thermal simulation earlier than phase B would not justify the expected insight. With this work we intend to improve the thermal design process by enabling the execution of automated thermal simulations in early design phases without spending much time on the detailed setup. Therefore, the workflow provides thermal engineers with expected spacecraft temperatures, to support the conceptualization of possible thermal control concepts.

The objective of this paper is to explore the possibilities of coupling MBSE tools to an environmental analysis tool, in this case thermal analysis. This is done by conceptualizing and implementing a round trip workflow that allows conducting automated thermal analyses using a system modeled in an MBSE tool, such as the open-source tool "Virtual Satellite". Precisely, this paper

1. shall evaluate if it is possible to create an engineering workflow that allows to perform automated thermal simulations in early project phases using data from an MBSE system model.
2. Furthermore, it shall be evaluated how well the automatically generated thermal model compares to a manually crafted identical model in order to validate the functionality of the workflow.

All tools required for the workflow should be available under open-source licenses.

2. State of the art

MBSE tools support different automated analyses. These analyses usually focus on functional aspects, interface compatibility, and size/mass or power budgets. Environmental aspects are not yet covered in most MBSE tools, although a first meaningful thermal analysis can be conducted based on relatively little information, compared to mechanical analyses for example. Especially the amount of geometric information required is quite small. Still, the activities of thermal engineers are often just limited trade-off studies, not using computational models in the earliest design phase [3]. More accurate design aids, such as thermal simulations, are employed at subsequent stages of a project. This is due to the fact that setting up a thermal simulation is time consuming. The geometric models and thermal properties from all subsystems, units, or components have to be collected and integrated in one complete model. Then, this composite thermal model has to be simplified and discretized, all boundary conditions have to be applied, and the simulation output has to be evaluated. All of these tasks are usually done manually. Consequently, it is not efficient to use these tools in early design phases where design changes are expected frequently and large parts of the simulation are rendered invalid.

The concept of conducting automated thermal analysis out of an MBSE system model was implemented in [5] before. However, there are several differences: One difference is that our work is based on a MBSE tool that is not focused on thermal aspects, but supports many other engineering disciplines with a comprehensive system model. Furthermore, this work focuses on the whole round-trip engineering from system model to simulation and back to the system model again, rather than a one-way simulation execution. Finally, our approach uses open-source tools only for executing the thermal simulation.

Another state-of-the-art approach for exchanging models and modeling data between various modeling and simulation tools is the functional mock-up interface (FMI). FMI is a tool independent standard for exchanging dynamic models and for co-simulation. Here the exchange is realized with a functional mock-up unit (FMU). An FMU is a zipped file containing the model description in a .xml file, the implementation in source or binary files and, if necessary, additional data or resources [6]. As the tools used for the workflow do not support FMI and have a legacy interface covering parts of the required interfaces already implemented, the approach was not considered in this work.

According to the International Council on Systems Engineering (INCOSE) definition MBSE is the formalized application of modeling to support the various design, development, verification and validation activities along all project or product life cycle phases [7]. The objective is getting away from document-based descriptions of requirements, dependencies, and interfaces and replacing them with formalized data models. Without intending this list to be exhaustive, at time of writing there are two main flavors of MBSE tools with fluent boundaries in between. Most tools are focusing on one of the two groups, but there are also tools that include parts of both groups.

The first group of tools focuses on a clear separation between needs definition and solution design in terms of logical/functional and physical system architecture. Capella [8,9] with its diagram-based architecture analysis and design integrated approach (ARCADIA) method is an example for this methodology.

Due to the goal of enabling quick simulations for MBSE tools, this paper concentrates on the second group.Tools belonging to the second group focus on a smooth and seamless integration of analysis/simulation results and iterative design updates to the digital model data as single source of truth. This concept can be denominated as "round trip engineering" and is also known as "data driven systems engineering" [10]. Examples are Valispace [11] and DLR's Virtual Satellite [12,13]. The latter is an open-source software supporting flexible extensions of the data model, the so-called conceptual data model (CDM). It is actively used in the DLR'sconcurrent engineering facilitys (CEFs) and new features and methods are regularly implemented as described in this paper.

3. Concept

With the ability to verify and validate design changes with an automated thermal simulation workflow, thermal engineers can focus more on the development of the thermal design, rather than the evaluation of recent design changes. This is particularly helpful in the fast-paced environment of the initial design phase. Furthermore, early analyses of the system can reveal design flaws that would inevitably cause significant time delays and cost increases, if spotted at a later stage of the project.

In order to address the question if it is possible to create such an engineering workflow a top-level concept has to be elaborated. There are multiple possible approaches to reach the goal of having a simulation result with the least possible manual effort. The basis of such a workflow is always an MBSE tool that contains the system model.

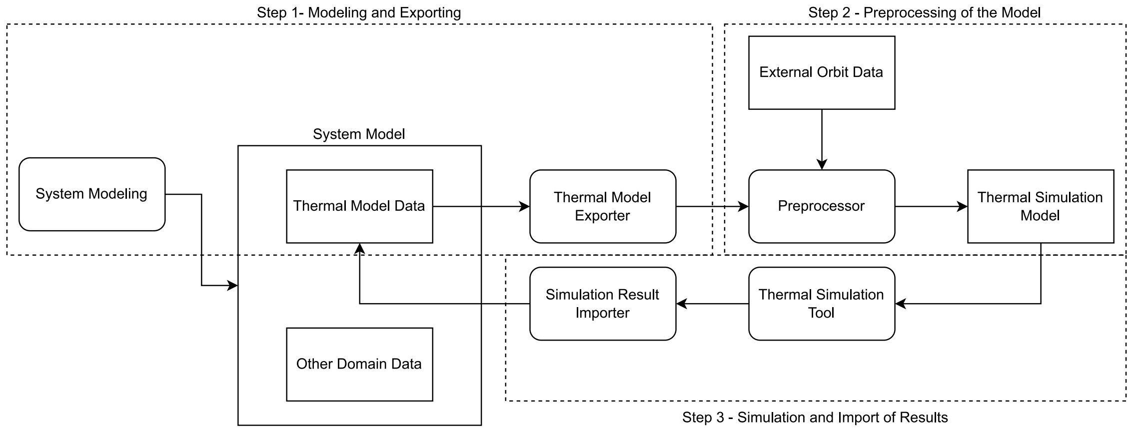

Fig. 1. Overview of the workflow concept.

The fastest approach would be to directly execute the simulation in the MBSE software with an integrated simulation tool. However, this is a very complex task and there are several drawbacks to this. The versions of the tools would always have to be maintained and it would not be possible to choose freely which simulation tool to use. Therefore, the preferred approach is to extract the thermal model from the system model and provide it to the dedicated thermal simulation tool after the necessary preprocessing required for the thermal simulation is done. The preprocessing can either be done using a preprocessor provided together with the solver, or, with a separate preprocessing tool that exports the thermal model in the correct format for the solver. Choosing a preprocessor independently of the simulation tool is another benefit this approach has compared to the fully integrated approach.

There are multiple ways to process the results after the simulation was executed. There is the option to visually check the results by engineers in a postprocessing tool, or in the result files provided by the simulation tool. More useful would be to have a loop going back to the MBSE tool, displaying key results from the simulation run directly in the system model. This requires an application able to automatically process the results and include them in the system model. This round-trip would then provide a quick feedback to the engineers, after the automated simulation process is initiated from the MBSE software.

Fig. 1 depicts the concrete top-level round-trip concept derived for the workflow. The flow of data displayed in the figure is described in the following. In order to execute a thermal simulation, a system model with its geometry and thermal properties is created in an MBSE tool. The thermal aspects of the system model are then exported as complete thermal model to an intermediate data format. This data is accessed by a preprocessing tool which prepares geometry and thermal properties. The following processing steps have to be applied:

- meshing the geometry,
- setting all boundary conditions,
- applying material and surface properties,
- defining contact conductivities, and
- apply simulation parameters (e. g., static/transient, time steps).

The geometrical shapes in MBSE tools are often modeled as complete predefined bodies such as cylinders or cuboids, which do not have individual faces or points. This leads to the problem, that the modeling of contacts between specific faces is not possible. This problem is solved by enabling the preprocessing tool to detect contacts and the involved faces on its own. Additionally, the components in contact with each other have to be defined together with their thermal contact conductivity in the system model. The defined contacts are exported together with the rest of the thermal model. The preprocessing tool then matches the detected contact faces with the contact bodies defined in the system model and applies the respective properties to them.

When analyzing the thermal behavior of space systems, it is also important to consider the thermal radiation acting on the object in orbit. For this purpose, the preprocessing was enhanced by the feature to include orbit data from an external mission analysis tool. The tool used for this can be freely chosen, the orbit data only has to be provided in a text file with a specific syntax, containing the information described in more detail in Section 5.2. This orbit data is then processed into resulting heat loads that are applied to the individual faces.

All this data related to the thermal simulation model is then written to a format that is accessible to the thermal simulation tool. The simulation tool executes the simulation based on this data and finishes by creating an output file. The output file has to be processed further to display the results of the thermal analysis in the system model. This concludes the analysis round trip by making the essence of the simulation results available to the project engineers via the system model.

4. Tools in the workflow

In this section, we introduce the open-source tools required for the automated thermal analysis. How the tools are utilized to implement the workflow is described in Section 5.

4.1. Virtual Satellite

Virtual Satellite is a model-based system engineering tool that is based on an extensible data model to support the whole life-cycle of a space system [14]. As shown in Fig. 2, its extension mechanism is based on the concept of engineering categories that can be typed by dynamic type definitions [1]. Virtual Satellite's language extension environment then generates Java beans for the engineering categories to simplify model access. Thereby, Virtual Satellite enables to execute "dynamic" Java applications, called Apps, that use generated code based on language extensions. Apps can be used to add, modify, and export data from the model.

Virtual Satellite was initially developed to support early-phase concurrent engineering sessions, such as in the CEF in Bremen [1]. However, its core development branch also targets later project phases. In contrast to the CEF version, the core version features product structures with multiple trees for configuration control and explicit modeling of equipment. For example, as shown in Fig. 2, a reaction wheel is modeled once in a product tree to specify its equipment properties. In a configuration tree, this reaction wheel is then instantiated for each occurrence on the actual system. Each instance can be configured with e.g., its position and – if necessary – equipment-specific properties, but inherits the properties from the product tree.

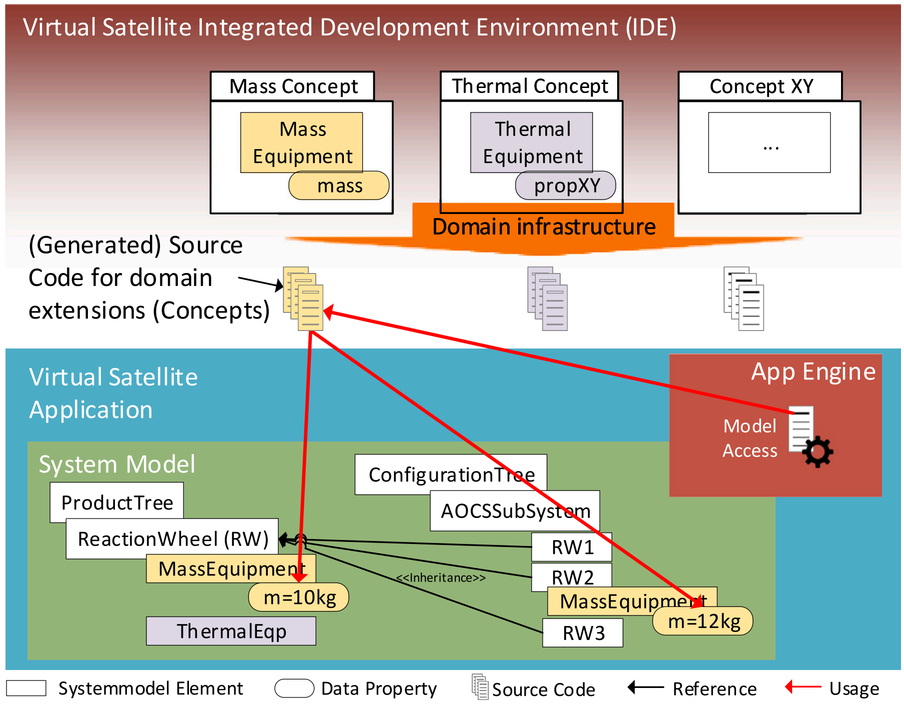

Fig. 2. Data model extensions defined in concepts are supported by generated Java bean classes that can be used by the App Engine to access the system data model.

4.2. FreeCAD

FreeCAD is a 3D CAD software, which is freely available under the open source license [15]. It is capable of all main tasks to be expected from a commercial CAD software and can easily be extended with so called workbenches. A workbench is a set of tools specially grouped for a certain task. There is a number of them included in the standard version of the tool. An existing Virtual Satellite workbench groups functionality to import and export data to the MBSE tool. It was originally created for importing a geometrical model from Virtual Satellite, changing the model's geometric features in a CAD program and afterwards re-exporting the changed geometry from the CAD program back to Virtual Satellite. The component structure, including the geometric information of the components, can be exported from Virtual Satellite in a JavaScript Object Notation (JSON) file. This file can be imported into FreeCAD using the before mentioned Virtual Satellite workbench. FreeCAD is also offering a FEM workbench and a Python console, which are used in the workflow.

4.3. CalculiX

CalculiX is a finite-element analysis tool that accomplishes thermal calculations [16]. It is a free and open source tool which was developed by Guido Dhondt and Klaus Wittig in the late 1990s and is still maintained and updated. The CalculiX application consists of CalculiX CrunchiX and CalculiX GraphiX. CalculiX CrunchiX is used to execute the simulations and calculations, while CalculiX GraphiX is capable of pre- and post-processing the simulation data, also providing a graphical user interface. Since the pre-processing in our workflow is done with Virtual Satellite and FreeCAD, CalculiX GraphiX is not required. Although the simulation results are parsed and fed back into Virtual Satellite, the post-processing of CalculiX GraphiX can be used to directly visualize it. CalculiX CrunchiX provides multiple solvers to handle linear and nonlinear mechanical, thermomechanical or thermal problems.

The format of the input data is similar to the input format of the commercial tool Abaqus. It is text based and consists of three main parts: The first one is the model geometry as a finite element mesh. FreeCAD can generate such a mesh from the geometrical model stored in Virtual Satellite. The second part is the thermal information, which is assigned to the geometry of the model. It consists of the initial temperature, contact conductivities, material properties, surface properties and boundary conditions. This information is extracted from the Virtual Satellite model and combined with the geometrical mesh from FreeCAD. The last part includes the simulation configuration. Here physical constants, the type of the simulation and the desired output needs to be defined. A FreeCAD script developed in this work provides the input files with the valid structure and syntax for the execution of the simulation.

5. Proof of the concept — implementation

The concept defined in Section 3 was implemented in the course of this work. For this purpose, the tools described in Sections 4.1, 4.2, and 4.3 were used. This section describes how the automated thermal simulation workflow was implemented.

The workflow is divided in three steps. The first step covers the required modeling of the system and the export of the model from Virtual Satellite. The second step processes the exported model in such a way, that it is available in the form of one or more valid input files for the simulation tool. The third and final step is the execution of the simulation and the feedback of the results to Virtual Satellite to close the loop.

5.1. Modeling and export in Virtual Satellite

In order to store the thermal model in Virtual Satellite the "thermal" concept was created. It extends the MBSE data model by thermal parameters and aspects of the simulation setup. This allows the user to store all parameters relevant for the thermal simulation in the Virtual Satellite system model. A list of the properties that are facilitated in this workflow is provided in Table 1, divided into properties that were already present in the data model and properties that were added with the "thermal" concept.

After modeling the system with its thermal properties, the export of the thermal model is done by two parallel tasks: The geometry and part

Table 1
Properties used for modeling the thermal model in this workflow. Two properties already existed before.

Property	Description	Virtual Satellite new/default
Element name and ID	Name and unique identifier of the element	Default
Element 3D geometry	The visualization of the geometry	Default
Element mesh size	Maximum characteristic mesh element edge length	New
Contact mesh size	Local maximum characteristic mesh element edge length for areas of contact	New
Thermal contact	Defines a contact between two elements	New
Thermal contact conductivity	Thermal contact conductivity of contact between two elements	New
Material: • Name • Thermal conductivity • Absorption coefficient (visible) • Emission coefficient (infrared) • Heat capacity • Density	Defines a material	New
Specific face properties: • Absorption coefficient (visible) • Emission coefficient (infrared)	Defines radiative properties for specific face	New
Analysis setup: • Analysis type (static/transient) • Time step • Total time • Include orbit radiation (yes/no)	Defines general properties for the thermal analysis	New
Boundary conditions: • Temperature (face) • Temperature (body) • Heat flux (face) • Heat flux (body)	Defines boundary conditions for individual elements	New
Results: • Maximum temperature (and respective time step number) • Minimum temperature (and respective time step number)	Shows the imported results of a simulation run	New

hierarchy is exported using the existing “CAD Export Wizard” export function, creating a JSON file containing the shapes from the model, the unique identifier of their respective component in the system model and the part hierarchy. The remaining part of the model is exported using the Java Apps developed in this work. They access the system model and write the thermal model data to text files. Most of these files have to be preprocessed further to serve as valid input for the simulation tool.

5.2. Processing in FreeCAD

The thermal model, which is now available in the form of the aforementioned JSON file and a series of text files, has to be processed further in order to correctly apply the boundary conditions and material properties. Furthermore, the geometrical shape needs to be meshed to be able to apply the boundary conditions and materials, and to be able to execute the simulation at all. The FreeCAD Virtual Satellite workbench is able to import the geometry file that was previously exported from Virtual Satellite. As mentioned in Section 4.2 the feature was already created for a different engineering round trip concept. However, in the application of this workflow, it is used to transfer the geometry described by the JSON file to an actual geometric representation in a CAD software. At this stage, the geometry could be exported as a regular Standard for the Exchange of Product Data (STEP) file to be further processed with any standard preprocessing tool.

The presented workflow implementation uses the preprocessing capabilities of the FreeCAD FEM workbench, meaning that the geometry is not exported but continued to be processed by FreeCAD. This has two advantages. Firstly, no additional tool is involved in the processing of the model, which keeps the complexity of the workflow execution, and therewith, the proneness to errors, low. Secondly, the FreeCAD Python console/API can be used to automate preprocessing by scripting the corresponding operations.

The FreeCAD console provides access to the geometric features of the model such as points, edges, and faces of a body. Furthermore, it allows access to the FEM workbench features, of which the integrated meshing tool “Gmsh” and the mesh exporter are the most important ones for the workflow.

A Python script that converts the complete model into a ready-to-use set of input files for the simulation tool CalculiX was developed for the presented workflow. The top-level tasks are listed in Table 2. The script first determines which faces actually are in contact with each other by facilitating FreeCAD geometrical operations. Since it is not possible to specify contact faces in Virtual Satellite, the automated preprocessing has to find the contacts itself. If a contact between bodies is either not defined in Virtual Satellite, or there is no actual physical contact detected in the preprocessing, it is neglected. Furthermore, geometric operations are performed to isolate the actual contact area from the whole corresponding geometric faces to refine the mesh only at the specific contact location.

After finding the contact faces, the geometry is meshed according to the mesh parameters defined in Virtual Satellite. The mesh is additionally refined locally at contact areas if specified in the contact definition in the system model.

As described in Section 3, in addition to the standard preprocessing tasks, the script is also capable of reading externally provided orbit data files, to calculate and apply the approximate orbit radiation influences as dynamic heat loads. The orbit data files must contain the time (step) information, the sun vector as seen from the spacecraft, the Earth vector as seen from the spacecraft, and the intensity of incident solar radiation as percentage of the maximum solar irradiation. The values can be exported from a mission analysis tool of choice. The script reads the data and calculates the actual heat load from direct solar irradiation, Earth albedo irradiation, and Earth infrared (IR) irradiation. In order to do this, the orientation of each individual mesh face has to be determined. Since at the time of the script development there

Table 2
Top level tasks executed by the FreeCAD preprocessing script.

Task	Description
Identification and definition of thermal contacts	The script identifies contacts between parts and (after meshing) applies the corresponding contact boundary condition with the conductivity value defined in Virtual Satellite.
Meshing	The script uses the Gmsh tool in FreeCAD to mesh the geometry according to the parameters specified in Virtual Satellite (including also local mesh refinement for contact areas) and exports the mesh to CalculiX input format.
Assignment of node/element sets	Mesh groups are created for the individual objects, faces and areas of contact in order to facilitate the application of boundary conditions and properties by grouping the corresponding nodes and elements into sets.
Application of properties/boundary conditions	Application and export of the material properties, heat fluxes, initial/fixed temperatures, and radiation properties to the corresponding node/element sets.
Calculation and application of additional heat fluxes from space radiation sources	Optional feature that calculates solar, Earth albedo, and Earth infrared heat fluxes from externally provided orbit data.

was no feature available in FreeCAD to unambiguously determine the orientation of mesh faces, this had to be implemented in the script by exploiting the predictability of the ordering of the nodes that form each element and their respective coordinates. In addition, the obstruction of each element face is determined by checking for intersections between the normal vector of a face and any other element. The script then calculates the resulting heat load factor for a face for each time step. Afterwards, the resulting factor is multiplied by the solar constant and applied to the face as face heat load. The same processing is done with the albedo and Earth infrared factors and their respective constants. The loads are then added up for each face and written into the input file for the heat flux boundary conditions.

5.3. Simulation and import of results

All generated input files are referenced in one main input file in their respective order. This main input file and all other input files referenced therein are then provided to CalculiX to execute the simulation. CalculiX generates a number of output files, one of which contains the information about all the temperatures at the different nodes at each time step. It also contains the heat fluxes to the FEM elements.

To close the loop of the workflow, the results have to be imported back into Virtual Satellite again. This is done by another Virtual Satellite application which reads the simulation output file, obtains the temperatures, and writes the maximum and minimum temperatures of each element of the spacecraft into newly created analysis results elements. This way, a quick check if the temperature exceeds the specified allowed operating temperature can be executed for each element.

6. Results

6.1. Validation

To enable the use of the workflow in space projects, it has to be verified that all Virtual Satellite exporters and Python script functions result in a valid model. For this purpose, the results obtained with them have to be checked for validity, meaning to check if the thermal model being created by the workflow matches the actual system model defined in Virtual Satellite.

The validation logic was to build a model in Virtual Satellite, process it with the workflow, and compare the simulation results with a model created manually in the graphical user interface (GUI) of a commercial thermal analysis tool. This way, all the automated preprocessing functions of the workflow would be tested against the preprocessing in a commercial tool. Since the provided workflow allows to do two different types of simulations, static and transient, at least two simulation models are required to check all functions. Of particular interest was to check the correct application of the contact boundary conditions for the thermal contacts between the components, since the automation of these conditions required by far the most effort during this work. The geometric model used for this validation is shown in Fig. 3.

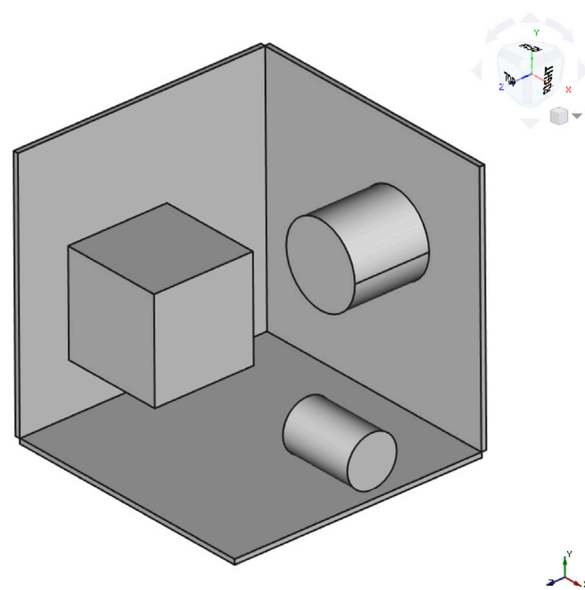

Fig. 3. Geometric representation of the validation model.

The obtained results showed that, as expected, the largest deviations occurred in the contact areas. The relatively coarse mesh, without local mesh refinement showed a maximum deviation between the automated simulation and the reference simulation of around 0.581 K. The maximum and minimum temperatures obtained for the whole model were around 314 K and 294 K, respectively. The deviation occurred close to the area with the highest temperature at a rather narrow contact area between a component with a volumetric heat flow boundary condition and one without any heat flow. Slightly increasing the number of nodes locally, by the means provided in the workflow, already decreased the

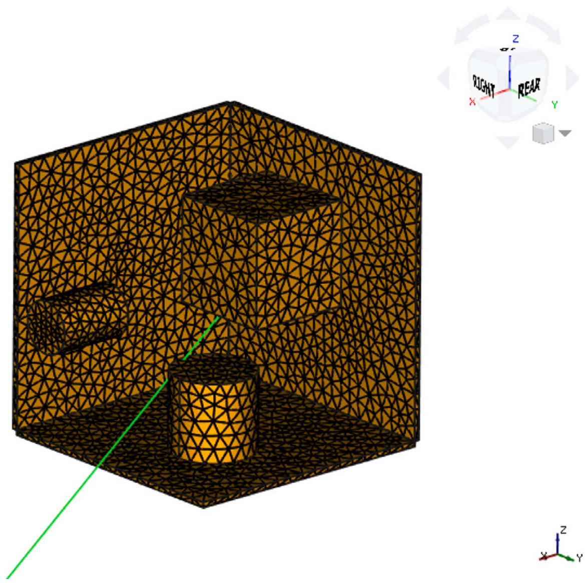

Fig. 4. Meshed validation model with green sun vector.

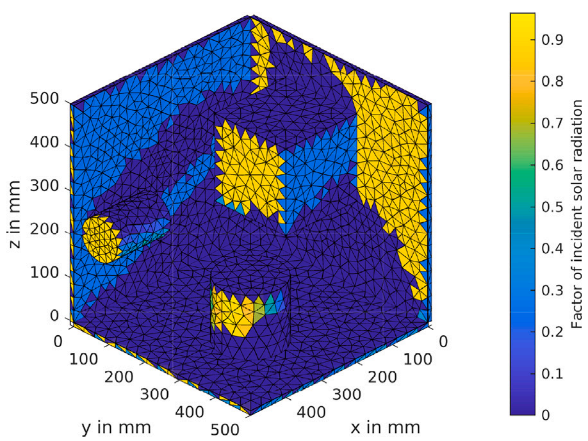

Fig. 5. Display of the solar irradiation intensity on each mesh face. Note: The visual representation is not accurate for faces next to body edges.

maximum deviation from 0.581 K (21,744 nodes) to 0.132 K (24,491 nodes), with the highest deviation occurring in the same region as before. The use of a further refined mesh reduced the maximum error to 0.084 K (34,815 nodes).

The result for the validation of the dynamic simulation was a maximum deviation of 0.052 K, even with the relatively coarse 24,491 nodes mesh. The validation of the orbit radiation feature could not be performed within the scope of this work. Still, the correct application of the corresponding heat flux boundary conditions was already part of the other models, and the visualization of the solar intensities on each mesh face shows good accordance with respect to the vector of the incoming light (see Figs. 4 and 5). A full validation of the workflow and all its functions requires more effort, which was not possible in the limited time frame.

6.2. Outlook

During the conceptualization and implementation of this workflow some aspects were discovered that would either improve the results or decrease the execution time. Since this work was mainly conducted as the main author's Master's Thesis with strict time limitations, these improvements could unfortunately not be applied. Nevertheless, some of the ideas are presented in the following.

6.2.1. FreeCAD Boolean function execution time

The execution of FreeCAD scripts is generally quite fast. Execution times of a couple of seconds for models with around 20,000 nodes were observed using standard office laptop hardware. However, when including the orbit radiation feature the execution time increases drastically depending on the size of the model. As a reason for that, a Boolean function of FreeCAD, used for determining the obstruction of a face, was identified. This function is quite slow and has to be executed several thousands of times, depending on the size of the model, leading to a significantly longer execution time compared to the execution without this specific function. Potential solutions would be to increase the efficiency of the function in FreeCAD, or to find a workaround that does not rely on this function.

6.2.2. Additional shapes

The amount of different shapes usable is currently limited, as Virtual Satellite only offers very basic geometric shapes. With the continuous development of the tool it might be possible to use more elaborated shapes in the future.

6.2.3. Automation of mesh analysis

As shown in Section 6.1, to ensure correct results, it is necessary to analyze the mesh. This mesh analysis could be done automatically by executing the workflow several times with increasing mesh sizes. The results could then be compared with each other to reflect the magnitude of the temperature deviations in the different simulation runs. As soon as no significant change of the resulting temperature is achieved with a finer mesh, the mesh parameters can be kept and used for subsequent simulation runs.

6.3. Result discussion

The obtained results show that the model set up by the automated workflow preprocessing is generally consistent with the manually created one. There are some deviations, but since two different tools were used this can be expected. Especially since differences in the treatment of contacts and heat flow boundary conditions were noticed in the validation process. Consequently, the region with the highest deviation is a region with a contact and a heat flow boundary condition applied to one of the contacting bodies. Still, even with a very coarse mesh the maximum deviation between nodes of the two models is relatively small, considering that the contact face where the deviations occur consists of less than 30 elements. Increasing the number of nodes locally leads to a decrease of the deviations, with a maximum deviation of 0.084 K for the models tested in this work. On the one hand, this indicates that the compared thermal models show a high degree of accordance. On the other hand, it also shows that the use of the mesh parameters provided in this workflow can be of high relevance for obtaining better results, especially close to contact areas.

Moreover, we could show that the workflow creates a valid thermal model out of the Virtual Satellite system model by means of the second simulation setup which employed a different component, with larger contact areas, as heat source. The maximum deviation occurring here is 0.052 K.

The results obtained in this validation are considered acceptable, since the deviations obtained even for very coarse meshes are relatively low compared to typical thermal design margins, especially considering the early stage of the design at which this workflow is intended to be used. Uncertainties in the range of 15 K are considered as typical for the thermal design in project phase A according to [3].

The tendency to built smaller satellites from commercial off-the-shelf components in the course of the "New Space" approach is in favor of this workflow. The thermal properties of these standardized components are well known and can therefore be modeled very easily in the early phases. In addition, there are efforts to provide databases with off-the-shelf component data to MBSE tools like Virtual Satellite in the future [17]. Selecting standard components from a database and automatically transferring their thermal properties into the system model would further facilitate execution of the workflow.

7. Conclusion

In Section 1 two research questions were posed with the intention to answer them in the paper. The first question can be answered with the obtained workflow. In this paper it is shown that it is indeed possible to execute automated thermal analyses in early project phases. The provided workflow allows to conduct thermal analyses without the need to spend much effort on the setup of the thermal simulation model. It allows to efficiently evaluate the impact of design changes with respect to thermal aspects. This is facilitated by scripting the entire preprocessing part, requiring the user to only execute the code developed in the course of this work.

This work also intends to motivate similar future developments in this area, which is stressed by the fact that only open source software is used for the execution of the workflow.

In addition, DLR's MBSE tool Virtual Satellite was extended by the capability to store thermal data. Thus, this work also contributed to the continuous development of the tool, moreover demonstrating that it is possible to participate in the development of Virtual Satellite without necessarily having to have a deep knowledge of software development.

The second question about how well the automatically crafted model compares to a manually crafted one is answered in Section 6. To summarize, the results obtained facilitating the provided automated preprocessing functions show good accordance with the results obtained by a similar, manual simulation setup with a different, commercial tool. There are small deviations though, which are considered to stem from the differences between the simulation tools, for example, the slightly different preprocessing of contacts and their conductance. When a sufficiently fine mesh is chosen the deviations are negligible, proving that the workflow fulfills its intention to automatically set up a valid simulation model.

The workflow was developed using the core version of Virtual Satellite. However, it was pointed out on multiple occasions that it could be used in the earliest phase of a project. Considering that there is a dedicated CEF version of Virtual Satellite it is of course of interest to extend the usage of the produced concept and code to the concurrent engineering phase. As the Virtual Satellite concurrent engineering facility's structure is very similar to the core version, only small adoptions of the code would be required. The general concept of the workflow could be adopted without alteration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] P.M. Fischer, M. Deshmukh, V. Maiwald, D. Quantius, A.M. Gomez, A. Gerndt, Conceptual data model: A foundation for successful concurrent engineering, Concurr. Eng. 26 (2018) 55–76.
[2] S.S. Cordero, T. Gateau, R. Vingerhoeds, MBSE challenges in the concurrent preliminary design of CubeSats: Nanospace study, in: IEEE International Systems Conference, SysCon, 2022.
[3] ECSS, Space Engineering – Thermal Analysis Handbook, ECSS-E-HB-31-03A, European Cooperation for Space Standardization, Noordwijk, The Netherlands, 2016.
[4] ECSS, Project Planning and Implementation, ECSS-M-ST-10C Rev.1, European Cooperation for Space Standardization, Noordwijk, The Netherlands, 2009.
[5] J. Gross, C. Messe, S. Rudolph, A model based thermal systems engineering approach, in: International Workshop on Systems and Concurrent Engineering for Space Applications, SECESA, 2012.
[6] A. Junghanns, T. Blochwitz, C. Bertsch, T. Sommer, K. Wernersson, A. Pillekeit, I. Zacharias, M. Blesken, P.R. Mai, K. Schuch, C. Schulze, C. Gomes, M. Najafi, The functional mock-up interface 3.0 - New features enabling new applications, in: International Modelica Conference, 2021.
[7] International Council on Systems Engineering (INCOSE), Model-based systems engineering (mbse) (glossary), 2022, https://sebokwiki.org/wiki/Category:Glossary_of_Terms. (Accessed 9 February 2022).
[8] P. Roques, Mbse with the arcadia method and the capella tool, in: 8th European Congress on Embedded Real Time Software and Systems, ERTS, 2016.
[9] Eclipse Foundation, Eclipse capella – open source solution for model-based systems engineering, 2022, https://www.eclipse.org/capella/. (Accessed 9 February 2022).
[10] P. Guardabasso, L. Lindblad, M. Witzmann, S. Siarov, Evaluation of the learning process of a data-driven systems engineering methodology in a workshop environment, in: 70th International Astronautical Congress, IAC, 2019.
[11] Valispace, Valispace website, 2022, https://www.valispace.com. (Accessed 9 February 2022).
[12] DLR Institute for Software Technology, Virtual satellite – the future of model based systems engineering for space missions, 2022, https://www.dlr.de/sc/en/desktopdefault.aspx/tabid-5135/8645_read-8374/. (Accessed 9 February 2022).
[13] DLR Institute for Software Technology, Virtual satellite, 2022, https://github.com/virtualsatellite. (Accessed 9 February 2022).
[14] P.M. Fischer, D. Luedtke, C. Lange, F.-C. Roshani, F. Dannemann, A. Gerndt, Implementing model-based systems engineering for the whole lifecycle of a spacecraft, CEAS Space J. 9 (2017) 351–365.
[15] The FreeCAD Team, Freecad – your own 3D parametric modeler, 2022, https://www.freecadweb.org/. (Accessed 9 February 2022).
[16] G. Dhondt, CalculiX CrunchiX User's Manual Version 2.16, 2019.
[17] D. Peters, P.M. Fischer, P.M. Schaefer, K. Opasjumruskit, A. Gerndt, Digital availability of product information for collaborative engineering of spacecraft, in: Cooperative Design, Visualization, and Engineering (CDVE), 2021.