GREATCUBE+: A MULTIDISCIPLINARY SOFTWARE TOOL FOR CUBESAT SYSTEM AND SUBSYSTEM DESIGN

Carlo Girardello[(1),(2)], Carsten Scharlemann[(1)], and Martin Tajmar[(2)]

[1]*University of Applied Sciences Wiener Neustadt, 2700 Wiener Neustadt, Austria.*
[2]*Institute of Aerospace Engineering, Technische Universität Dresden, 01062 Dresden, Germany.*

April 8, 2022

Abstract

CubeSats development is an iterative process that requires time and expertise. To guide CubeSat groups, specifically during the conceptual phase, the system engineering tool GREATCUBE+ has been developed. Its current version is capable of performing the conceptual design for CubeSats in the range of 0.5-30 kg with a maximum form factor of 12U. GREATCUBE+ considers multiple scenarios and allows the user to select specifically what type of payload, power, alignment and total data are planned to be implemented. GREATCUBE+ will be structured in three consecutive layers: an empirical, an analytical and a numerical level. The empirical level as starting point takes the initial top level inputs such as payload weight and power needs etc. which are given by the user and it returns design solutions based on empirical data. The results of this layer are associated to COTS with high TRL. The analytical level processes these results. The design refinement will proceed with well-established analytical equations following a predefined analytical path. The outcome of the refinement level is an optimized setup of COTS together with values for CubeSat mass, form factor, power production together with specifications of the single subsystems. A validation is obtained by a comparison with known CubeSat missions.

1 Introduction

CubeSats have been proposed for the very first time in the late '90s by Professor Puig Suari and Bob Swiggs from CALTEC and Standford University. They are intended to provide students a "hands-on" experience on assembling, building and designing a satellite. It is motivated by the fact that large size satellites are expensive and time demanding [2]. CubeSats then became a worldwide phenomenon and nowadays are widely used by universities as educational tool to industry. areAccording to the CubeSat Standards, Rev. 13, a CubeSat size is measured in U (Units) where 1U is equal to a cube 10cm x 10cm x 10cm and its maximum weight should be less than 1.33kg per U

Nowadays there are more then thousands CubeSats already in orbit. Until the first of January 2022 more than 1500 CubeSats and more than 1800 Nanosats have been launched, including two interplanetary CubeSats [7]. A large amount of industrial companies are involved in CubeSat technologies, with more than 500 factories worldwide producing and testing CubeSat components. One of the main downsides related to this type of satellites is the elevated failure rate. Roughly 35% of educational CubeSats and 25% of CubeSats developed by industries do not survive either launch nor the first few days of operations[12] [11]. The reasons for those numbers are the extensive usage of non space-rated COTS components, low amount of testing and inconsistencies in design [11]. To increase CubeSats success rate, this paper proposes a software tool which applies multiple methods to suggest to the user a complete CubeSat design at system level (weight, form factor, power production), as well as a choice of COTS subsystems and a step file of the CubeSat model assembled.

2 GREATCUBE+

GREATCUBE+ (hiGh eneRgy dEnsity cubesAT Conceptual User Based Engineering +tool), or GC+, is a software tool, which is currently under development at TU Dresden and FHWN. Its core characteristic is flexibility. As a software for CubeSat design, it must cope with a large variety of potential payload types, in order to provide a useful and universal method for future CubeSat missions. A Concurrent Engineering (CE) methodology ensures an optimal performance for complex systems with multiple interconnections. Additionally, CE has been established in the development of space systems by academia, industry and space agencies and it can be considered a valid and useful methodology. It includes the user as a customer, not as a designer, which is perfectly applicable in a Phases 0/A scenario where a limited amount of detailed technical information is available. By using GREATCUBE+, CubeSat teams can benefit from a time reduction in the conceptual development phase and can focus more on testing. The software is divided in three layers: An empirical, an analytical and a numerical one (Fig. 1).

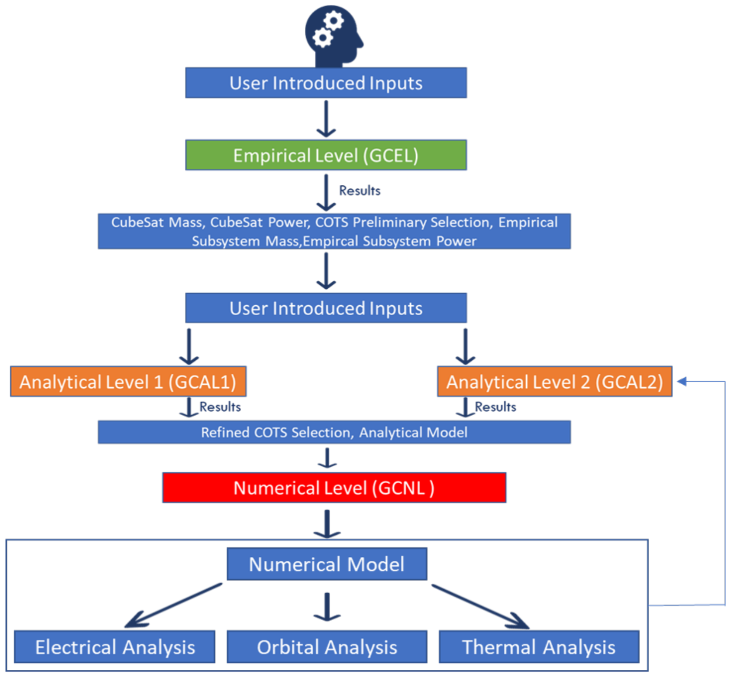

Figure 1: The architecture of the GREATCUBE+ software tool

The first interaction of the user with the tool is the introduction of the required basic inputs. They are divided into mandatory and not mandatory inputs. Further details can be found in Section 4. Simplified, those inputs serve to retrieve system information from empirical relations retrieved from an internally collected database of published CubeSat designs. In addition, the same relations are used to indicate the configuration of the internal components at subsystem level. The final result is a preliminary CubeSat model with system and subsystem level information. At this point, if desired, the simulation can be stopped by the user. In the second, analytical,level (Fig. 1), GC+ utilizes the empirical preliminary model to perform an analytical refinement of the subsystems comprising the CubeSat. The two separate analytical models (GCAL1 & 2) are explained in Section 5. The refinement uses well established analytical equations, which have been published by the scientific community [13, 5], to select the best performing COTS components. In this level, some additional mandatory input information is required and used to perform a correct run of the analytical level. They are: payload volume, payload peak and idle power consumption, payload temperature range, time of required alignment per orbit, duty cycles of the other subsystems, altitude, expected duration of the mission and radiation hardness of the payload. With those inputs it is possible to perform an optimization of the COTS components suggested in the empirical level by using the analytical results (e.g a selection of the proper ADCS after having computed the maximum amount of torques the satellite will suffer). The model obtained at the end of the analytical level will be accurate at both system and subsystem level. In addition,it includes a detailed list of recommended COTS components for the user together with a list of justifications for those specific selections. At this point, if necessary, the simulation can be stopped by the user. In the numerical level, the information inherited from the analytical level will serve to automatize a CAD assembly. With this model an orbital, thermal and electrical simulation will be performed. The final layer will provide numerical confirmation of the proposed setup or it will allow the user to identify the critical points of the design.

3 Methodology

The methodology used for the development of GREATCUBE+ is Concurrent Engineering (CE). This specific methodology has been selected based on a comprehensive literature review of possible approaches utilized in different satellite projects for specific types of payloads. CubeSats are complex systems with many interconnections between subsystems and in the conceptual phase, developers usually need to perform multiple iterations to reach an optimized result. This process of back and forth is time intense - additionally to being underperforming - and may lead to hidden inconsistencies, which might cause the loss of the mission [11]. The main trade off was done between several approaches of space teams in the development of their missions (sequential engineering, concurrent engineering, collaborative engineering, simultaneous engineering) together with comparable software tools from literature [4, 3, 9, 1, 6, 8] and many more. It results in the identification of the Concurrent Engineering methodology as the optimal approach. The main reason for the implementation of CE arises from the necessity of a method which could treat the simultaneous development of a large complex system with multiple internal interconnections. Further details can be found in Section 5. In addition, Concurrent Engineering is commonly accepted in academia and industry via Concurrent Engineering Challenges. Summarizing, these challenges take place over several days, from which the first day is usually dedicated to the specification of the mission requirements. Sequentially, subsystem experts perform an initial trade-off in the area of their expertise and finally the results are collected and stored for usage on the following day by the System Engineer. Within the second day, each expert proceeds with the new model to perform additional refinements on its own field of expertise considering constraints, imposed by the remaining subsystems within the previous iteration. The process concludes with a final design, that is assembled and proposed to the committee at the end of the Challenge [10].

4 GREATCUBE+ Empirical Level: GCEL

The first level, called GCEL (GREATCUBE+ Empirical Level), is the main engine of GREATCUBE+ and will be discussed in this section. It employs a set of internally collected empirical correlations retrieved from an in-house built database of satellite missions, which correlate various parameters to each other. An example is the bond between CubeSat Mass and Form Factor. In figure 2 a schematic of the processes performed by GCEL is illustrated.

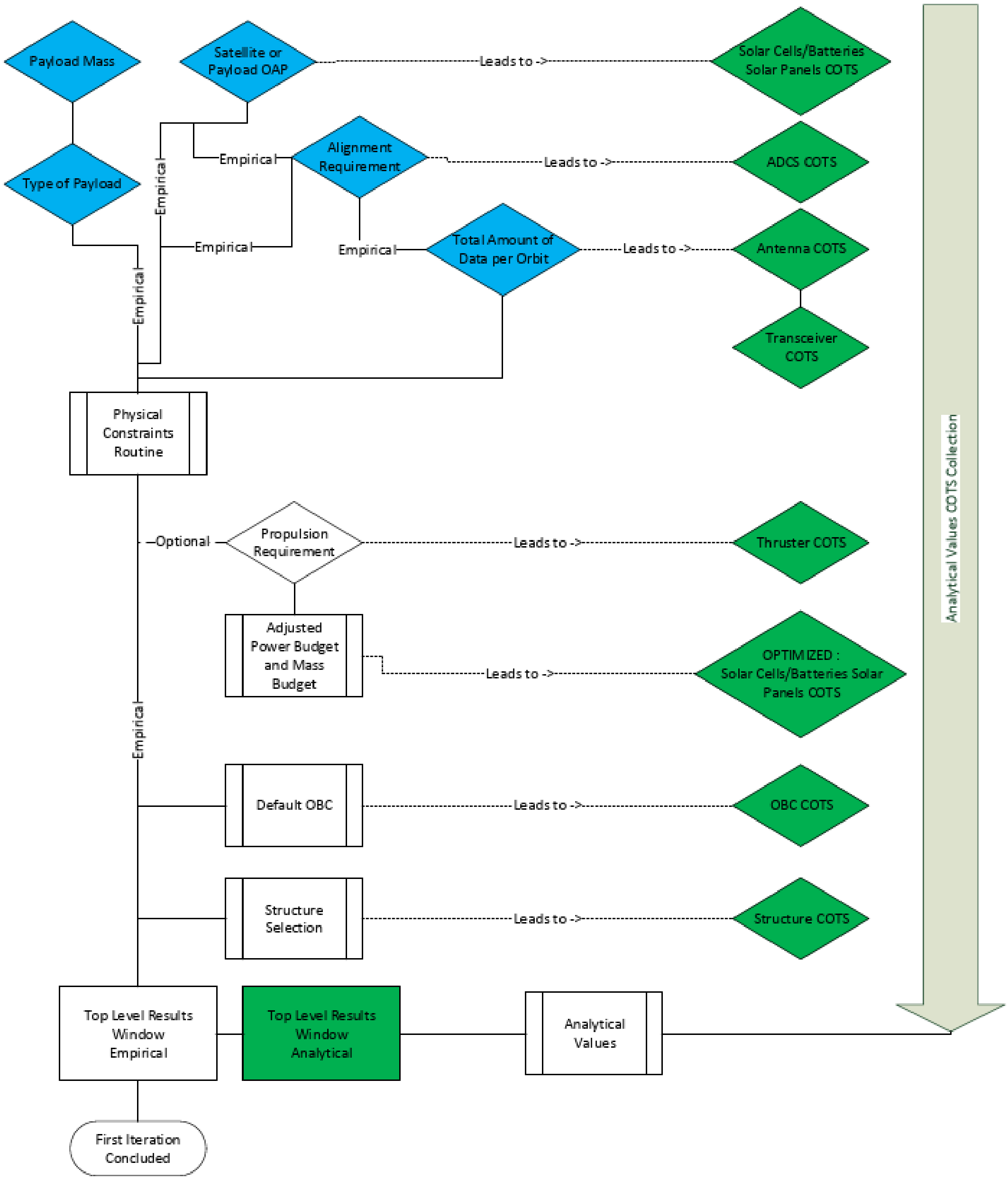

Figure 2: GCEL Architecture

The mandatory inputs of GCEL are Payload Mass, Satellite On Orbit Average Power (OAP) or Payload OAP or at least one of them. Once the Satellite or Payload OAP have been introduced, the user has also the chance of providing the alignment requirement and the total amount of data to be transmitted, which will be sent to a user-specified number of ground stations (GS). GCEL can work with any number of the previously introduced inputs but some constraints are applied. If, as an example, a propulsion system is the main payload, it will not be possible to proceed until an alignment requirement and the payload or satellite OAP have been specified. Once all the parameters have been set in the GUI (Fig. 3), the user shall click on "Run Simulation". The software tool guides the user upon the steps, necessary to run the simulation.

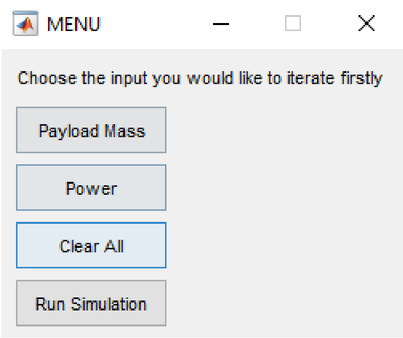

Figure 3: GREATCUBE+ Empirical Level GUI

Sequentially, the tool starts to compile all the separate scripts of the empirical level, which are containing the empirical laws for correlations. Using these correlations, it is possible to create a chain of equations which link the payload mass to the CubeSat mass, to the form factor and, finally, to an experimental power consumption. This process is performed for each introduced input. There are four main decision tree scripts:

- **Payload Mass Starting Point Decision Tree:** Starting from the payload mass input, the code retrieves the CubeSat mass, the satellite OAP and the form factor of similar CubeSats.
- **Satellite/Payload OAP Starting Point Decision Tree:** Starting from the satellite/payload OAP input, the code retrieves the CubeSat mass and the form factor of similar CubeSats.
- **Alignment Starting Point Decision Tree:** Starting from the satellite/payload OAP input, the introduction of the alignment requirement ensures that it is possible to estimate the ADCS mass of comparable alignment systems. From this value, the respective empirical CubeSat mass, the form factor and the satellite OAP are retrieved.
- **Total Amount of Data Starting Point Decision Tree:** When the Alignment is introduced, the link-budget decision tree is needed. With this, together with the number of available ground stations, the TT&C power requirement is estimated. This result is required to simulate the satellite OAP, the CubeSat mass and the form factor from empirical correlations.

Finally, GC+ processes all the outputs of the different decision trees via internally introduced weights on the results. The weight's values are associated to the R^2 values of the fits on empirical data. The weights are necessary to consider the different inputs of the equations used to calculate specific intermediate results. An example is the CubeSat mass vs. payload mass and CubeSat mass vs. satellite OAP plot. Satellite OAP and payload mass are inputs introduced by the user. The final output provided to the user is the CubeSat mass but the empirical relation to calculate it has a different fitting with another R^2 value. Already at this stage, each time an empirical correlation can be identified between subsystem components and main system components (e.g. ADCS mass vs. CubeSat mass based on the accuracy of the alignment) the software automatically selects a COTS component with those features from a database of components. That specific component will be used as a reference for the subsequent steps in the analytical level. Once the results have been weighted the code runs autonomously a routine (Physical Constraints Routine, PCR) over the consistency of the given outcomes in terms of form factor and power budget. The PCR serves to verify the consistency of the final weighted values of the empirical trees with the single values, which will be subsequently used. At this stage, the empirical correlations have already defined a model in terms of CubeSat mass, volume and power production. Once the PCR has performed its verification, the model proceeds with the Propulsion Requirement. A propulsion subsystem is a relative exotic feature for CubeSats and its possible implementation alters the consistency of the empirically retrieved outputs collected so far. The reason for this statement is that a propulsion unit, implemented in a 3U CubeSat as an example, may change its weight and volume up to 33%. If a thruster is employed, the software adds it to the empirical values of CubeSat mass and power consumption and it runs a crosscheck of the power subsystem architecture. Sequentially, the COTS OBCs and structure COTS subsystems are as well calculated based on the empirical model to be considered in a separate hybrid (empirical and partially analytical) model. Finally, GCEL has reached its last stage and two different models were simulated. The first is covering the system level characteristics of mass, power and form factor together with empirically retrieved values of COTS components, and one which is fully composed of real COTS products, which have been matched to the empirical values. In order to estimate the most accurate model, a set of eight known CubeSat missions has been collected and fed to GCEL. There is a trend, that the empirical values are a better representative of the system level characteristics. These values, together with the suggested list of COTS components, are fed to the following analytical layer, if the user is satisfied with the results, and proceeds with the simulation. It is worth noticing that on the right part of figure 2, the set of proposed COTS products are constantly linked to the empirical analysis.

5 GREATCUBE+ Analytical Level

The Analytical Level is the second layer of GREATCUBE+. Its input is the model obtained in GCEL, and it performs an analytical fit of the subsystem components in order to verify if the COTS specifications suggested in GCEL fit to the spacecraft analytically. The goals of the analytical level are to increase the level of detail of the model and to provide an analytical motivation for the COTS product choices. The final results of GCAL are a summary of the new updated recommendations for COTS based on analytical laws, inherited from [5] and [13], together with an updated CubeSat model.

At this level, some extra information from the user is required for running the simulation. Those necessary and mandatory, inputs are: payload volume, payload peak and idle power consumption, necessary time for the alignment requirement, duty cycles, minimum and maximum payload survival temperatures, altitude and thruster misalignment. Reasonably, if the payload or satellite OAP were not introduced in the empirical level, the data relative to peak and idle power mode linked to the payload are not requested at this stage. The same applies for thruster misalignment in case a propulsion system would not be present. Those user introduced inputs are necessary to perform a detailed study of the subsystems. As an example the CubeSat mass, the form factor and the solar panel configuration, inherited by the GCEL are used to calculate the inertia matrix of the spacecraft. The altitude and the alignment time serve to compute the orbital period and the necessary amount of torque which needs to be stored in the system. This is sequentially applied to estimate the various external environmental torques acting on the spacecraft, which will result in a total disturbances estimation over the period of one single orbit. This analytically retrieved result enables the selection of a proper ADCS, which will ensure the correct attitude control of the CubeSat. In the case this result differs to the GCEL COTS component selection, the latter will be substituted. The analytical level introduces various refinements. In particular, they are: Telemetry Telecommunication and Command (TT&C) refinement, On Board Data Handling refinement (OBDH), Attitude Dynamics and Control Subsystem (ADCS) refinement, Electric Power Subsystem (EPS) refinement, Structure Subsystem (STR) refinement, Thermal Control Subsystem (TCS) refinement, Payload refinement (PL) and Propulsion refinement (Prop Ref). The analytical level follows the principles of Concurrent Engineering. For this reason it is mandatory to analyze how the data are managed internally. The questions to be addressed by the development of two separate models are:

- Should the subsystems have a specific order of refinement?
- Should all the subsystems perform their refinement starting from the same model?
- If a new COTS component is recommended, does it substitute the one present in the current model or should it be crosschecked at the end of the simulation?

These are the reasons, which entailed the development of GREATCUBE+ Analytical Level 1 and 2 (GCAL 1 and GCAL2). In the current version of GREATCUBE+, both models are simulated, in the final version, only one of them will be selected.

5.1 GREATCUBE+ Analytical Level 1: GCAL1

GREATCUBE+ Analytical Level 1 (GCAL1) operates with the approach introduced in Section 5. It processes the user introduced inputs and the inherited GCEL model to perform a refinement of the COTS components suggested which leads to a more accurate CubeSat model at the end of the process. The workflow and the working principles of this model are illustrated in figure 4.

The access of each subsystem to the same initial model results in a mono-directional flow of information. Additionally, there is no a priori specification to any type of subsystem order of refinement. An individual refinement is subsequently performed over single COTS products, which are merged together in the Analytical Model Assembly (AMA). Finally, the AMA goes through a reality check to be passed and proceed to the numerical level or - in case of failure - to be used as a new starting model for an additional iteration. The reality check only requires the AMA to be performed. It scans the set of COTS suggested in the AMA in terms of volumes, masses and power requirements to detect discrepancies (e.g. 1U with 12 solar panels). If the check does

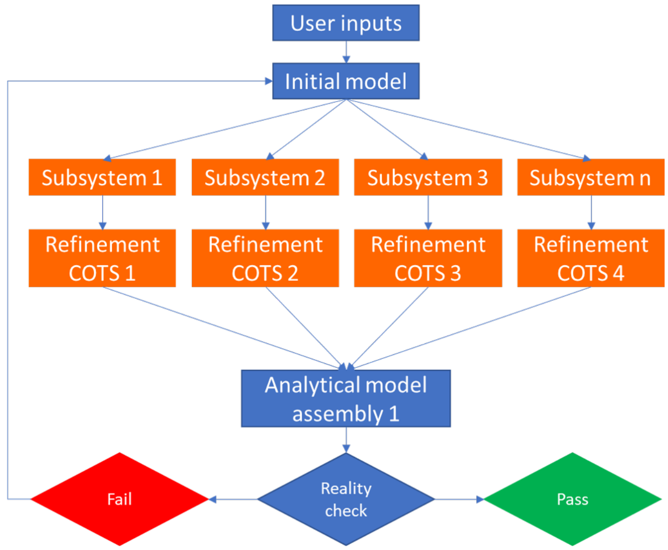

Figure 4: The workflow of the GREATCUBE+ Analytical Level 1

not identify any flaws, the AMA is validated and allows to proceed with the Numerical Level. If, instead, an inconsistency is detected, GCAL2 performs an additional refinement utilizing a bigger, or smaller, form factor and a new COTS selection is provided. The core concept of GCAL1 though is the absence of any prioritization of subsystems, motivated by the Concurrent Engineering Challenge description introduced in Section 3. GCAL1, in a real concurrent engineering facility, takes place after the first step, where an initial model is assembled and presented for further refinement to the experts. Utilizing this approach, the risk of biased results due to personal, subjective decisions from the system engineer are avoided.

5.2 GCAL2

GREATCUBE+ Analytical Level 2 is the alternative method developed to answer the questions stated in Section 5. Like GCAL1, it will process the user inputs and the inherited GCEL model to perform a refinement of the suggested COTS components, which will lead to a more accurate CubeSat model. A flow diagram of the tasks performed during a normal run of GCAL2 is illustrated in figure 5.

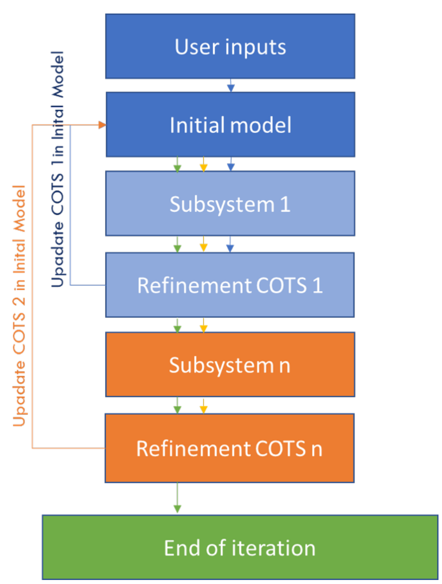

Figure 5: The workflow of the GREATCUBE+ Analytical Level 2

The differences to GCAL1 is the non mono-directional flow of information and the introduced prioritization of subsystems. As a real life aspect, this method takes into account, that the system engineer is experienced on subsystem design and provides an in depth knowledge of the critical systems. In this case, the project will ideally take less amount of time since the system engineer knows already, which subsystems need more iterations. The methodology of this process is still Concurrent Engineering but with an intended bias over the subjectively important subsystems. After the user introduced inputs, the first subsystem performs its refinement utilizing the GCEL model and a new optimized COTS product is identified. Sequentially, this new COTS component substitutes the one proposed in the GCEL model. Finally, GCAL2 checks if volume-, mass- and power-wise the new subsystem can fit into the current setup. If it does not, the size of the CubeSat is increased, or decreased, and a new iteration on the subsystem identification is performed, until convergence is reached. Once it is steady, the optimization of the next, less critical subsystem is triggered on the updated model.

6 GREATCUBE+ Numerical Level: GCNL

The Numerical level of GREATCUBE+ (GCNL) is the third and final layer. It collects the information inherited from the Analytical Level and it creates a CAD-file to be analyzed in thermal, orbital and electrical simulations. This level, initially, will create blocks representing the subsystems with the characteristics inherited from the producers data sheet, which come together with the COTS component list at the end of the analytical level. In this way, each block will have its volume, internal power consumption, material, operating temperatures and voltage requirement. With all this information, an assembly of a 3D model is created and the file is then moved to an external software for the simulation. The numerical results of the analysis performed so far by the GREATCUBE+, will be then provided to the user.

7 Results

Currently, the development of GC+ recently concluded the two analytical levels mentioned in Subsections 5.1 and 5.2. In the current version GREATCUBE+, is composed of GCEL, GCAL1 and GCAL2. The software has been validated throughout the whole development process as presented in this section. In order to define the boundary conditions, a few remarks concerning the expected outcomes are necessary. Firstly, the validation of the software is done by introducing the parameters, which are representatives of successful CubeSat missions in GCEL and GCAL1, 2. Secondly, the results are considered acceptable in terms of system level outcomes (CubeSat Mass, CubeSat Form Factor and CubeSat Power Production) if they stay in a 10% range, compared to the empirical values. Thirdly, in case some of the inputs required from GREATCUBE+ to run are not available, they are assumed and listed. Fourthly, if the results are above the expected 10% threshold it may be because of the components composing the real CubeSat are not present in the current database. The code has been validated eight known CubeSats respectively for GCEL and GCAL1, 2. The results of the iterations performed on those CubeSats for the Empirical Level are presented in figure 6 for what concerns the CubeSat Mass.

The average empirical error for the total mass is approximately 23.3%, while it is about 12% for the CubeSat form factor. Its notable that the introduction of a few simple inputs, commonly available at the beginning of every mission conceptual phase, enable the identification of roughly the total CubeSat mass and the overall form factor already at the empirical level. In addition, the final model of GCEL includes a recommendation on ideal COTS components, as explained in Section 4. Sequentially, these values are fed to the analytical level in addition to the necessary inputs. In the vast majority of cases some of the data could not be retrieved since they have not been found to the authors best knowledge, so they needed to be assumed and retrieved from comparable missions. The most common data which could not be found are the duty cycles together with payload volumes. If not provided, a reasonable assumption for the duty cycles was: 35% of the orbital period the payload is operating at peak power mode and for the 65% of the time in idle power mode, 10% peak power mode for ADCS and Propulsion, 100% peak power for OBC, 25% peak power for TT&C. If the payload volume was not listed, the assumed value varied based on the type of payload according to the analyzed internally collected database (e.g. a 0.15kg payload and its architecture occupy approximately 0.15U based on empirical data). The results of the above introduced analysis are presented in Figure 7. In the case of GCAL2, no information for any CubeSat has been provided related to which subsystem was the most critical, a standard approach which considers the following order was implemented: Payload, ADCS, OBC, TT&C, Structure, Thermal, Propulsion, EPS has been considered. In cases, like CLIMB and NANOBOB, in which many of the required information was available, the accuracy of the results was well below the threshold limit of 10% for the final CubeSat mass and power consumption for both GCAL1 and GCAL2 compared to the final values with the literature ones. If a large amount of assumptions was necessary due to the lack of data, like in TEMPEST-D, the results of GCAL1 and GCAL2 did not surpass the accuracy obtained in the empirical level, approximately 20%. In order to provide a more comprehensive explanation of the capabilities of GREATCUBE+ with all necessary information available, the CLIMB mission, which is currently under development at the University of Applied Sciences Wiener Neustadt will be analysed. CLIMB is a 3U CubeSat, which has the goal of lifting its apogee up to the altitude of the Van Allen belt to take measurements of the variations of Earth's magnetic field and additionally to test out the radiation hardness of its hardware. It will increase its orbit sequentially via a FEEP thruster (Enpulsion NANO). It will include eight

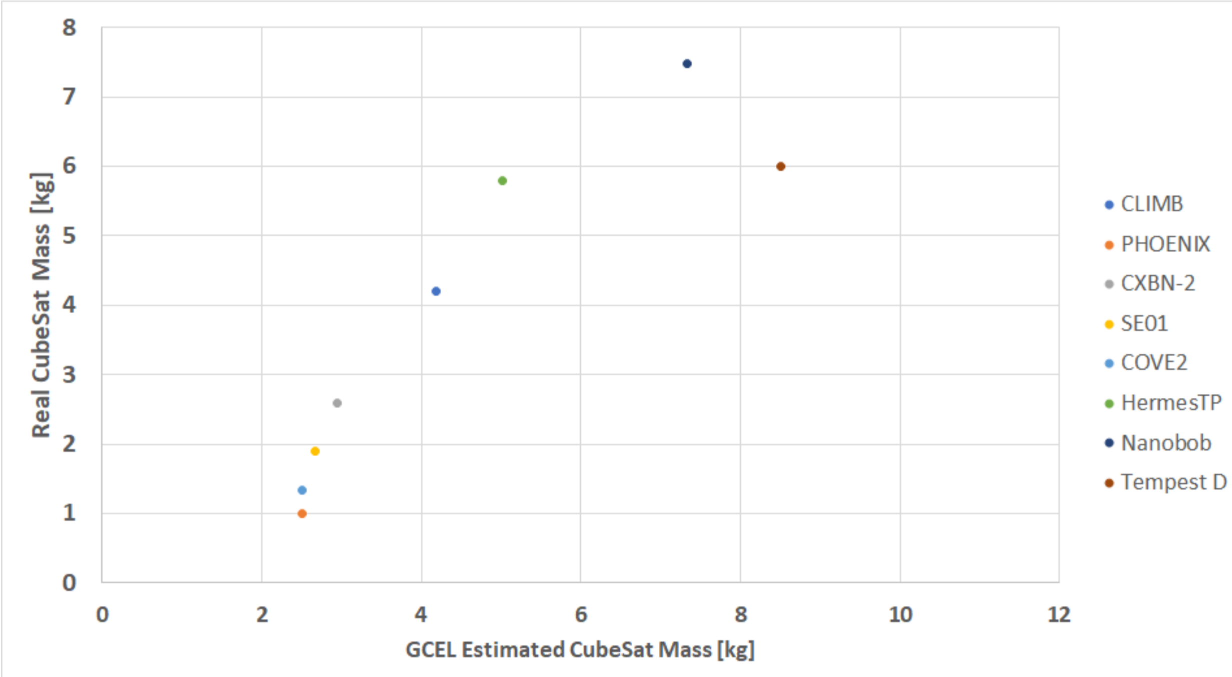

Figure 6: Validation of the GREATCUBE+ Empirical Level w.r.t. CubeSat mass

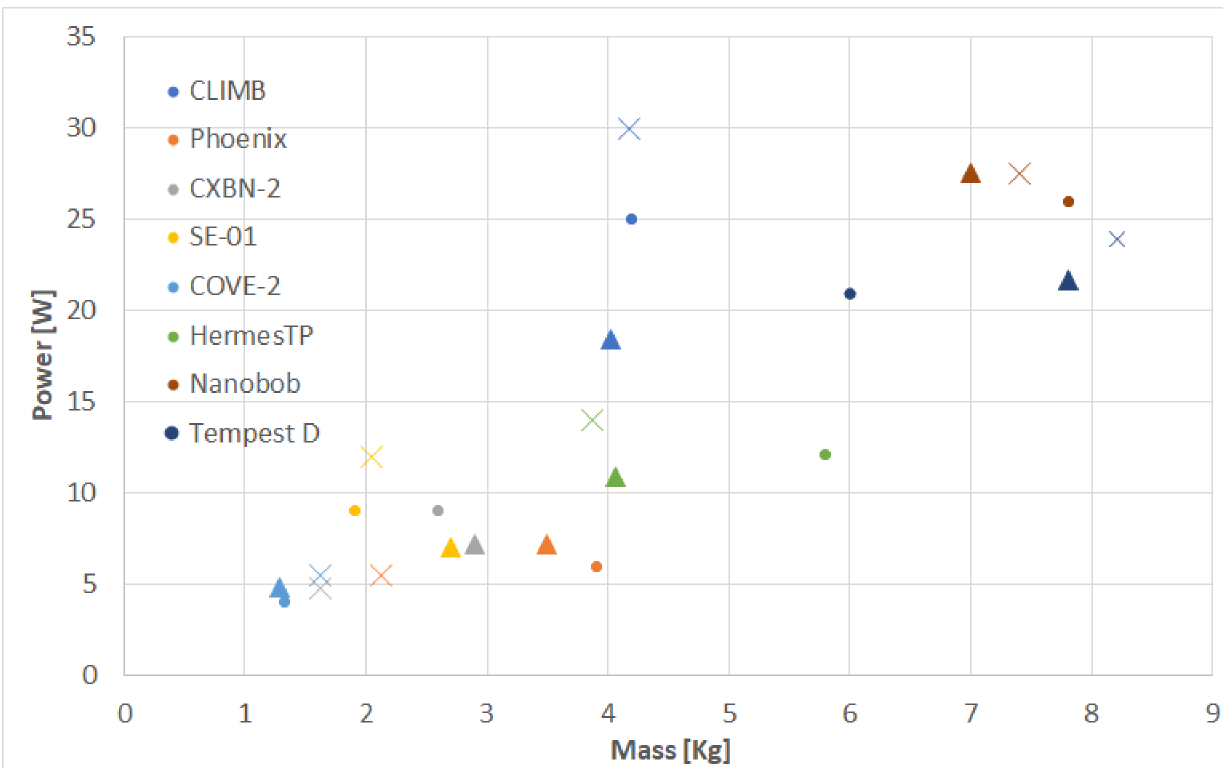

Figure 7: Results of GREATCUBE+ Analytical Level 1 and 2 Results. GCAL1 results are marked with triangles, GCAL2 results are marked with an X and literature values are marked with dots.

lithium ion batteries and it will use an S-Band transceiver for telecommunication. Its payload is a magnetometer with weights 0.016kg, consumes on average 3W, 5W during peak mode and 2 W during idle mode for 35% of its orbital time. Its propulsion system and ADCS will operate together for approximately 10% of the orbital period in peak mode, while the OBC will be 100% in peak mode and the TT%C only for 25% of the time. The altitude considered for the simulation was of 600km. The

Table 1: GCEL: Empirical Results, CLIMB case

	GCEL Value	Real Value
CubeSat Mass	**4.3**	**4.2**
Form Factor	**3**	**3**
Power Production	**17**	**25**
Solar Panels Nr.	**2**	**5**
ADCS	**Closed Loop ADCS NewSpaceSystems**	**TBD**
Propulsion	**Enpulsion Nano R3**	**Enpulsion Nano**
TT&C Type	**VHF GomSpace**	**S-Band In-House built**
Structure	**3U ISIS Space**	**In-House built**
OBC	**ISIS OBC**	**TBD**
Battery Pack	**6 Li-Ion Batteries**	**8 Li-Ion Batteries**

empirical level, GCEL, results of those inputs are presented in Table 1:

The numerical results are quite close to the literature values of the satellite already at this empirical level. Concerning the COTS product selection, the values which caused the suggestions proposed in table 1 are retrieved from empirical correlations but they are quality-wise similar to the ones which will be implemented in CLIMB. This model was satisfactory, so it has been decided to proceed to the analytical levels (Tab. 2). For this simulation, both the analytical methods (GCAL1 and GCAL2) have been considered. The necessary inputs are listed in the previous sentences. For the GCAL2 method, the following subjective iterative routine has been implemented: Payload, ADCS, OBC, TT&C, Propulsion, Thermal, EPS.

Table 2: GCAL1 and GCAL2 Results, CLIMB case 400 km

	GCAL1 Value	GCAL2 Value	Real Value
CubeSat Mass	**4.15kg**	**4.2kg**	**4.2kg**
Form Factor	**3U**	**3U**	**3U**
Power Production	**30W**	**30W**	**25W**
Solar Panels Nr.	**5**	**5**	**5**
ADCS	**Closed Loop ADCS Y Mom NewSpace Systems**	**Closed Loop ADCS Y Mom NewSpace Systems**	**TBD**
Propulsion	**Enpulsion Nano**	**Enpulsion Nano**	**Enpulsion Nano**
TT&C Type	**VHF GomSpace**	**VHF GomSpace**	**S-Band In-House built**
Structure	**3U ISIS Space**	**3U ISIS Space**	**In-House built**
OBC	**ISIS OBC**	**ISIS OBC**	**TBD**
Battery Pack	**6 Li-Ion Batteries**	**6 Li-Ion Batteries**	**8 Li-Ion Batteries**

The results are matching for both GCAL1 and GCAL2 in terms of COTS products suggested for the subsystems and the same applies for the system values (the 50g difference is caused by a different selection of solar cells). The average error, in the case all the necessary information have been available, is close to 4%. Additionally, the current simulation has been performed for a 600km circular orbit. Based on the results obtained by increasing manually the orbital altitude and keeping track of the provided results, for a final orbit like the one CLIMB targets to achieve (1000km) the results would be, as presented in Table 3:

It shall be noted that the battery pack difference arises because GCAL1 and GCAL2 are not considering the damages caused by constant charging and discharging cycles if low DOD are reached, which degrades the batteries. This was instead taken into account by CLIMB team. In order to solve this issue, in addition to the already present lifetime degradation, a decay factor due to the batteries charging and discharging cycles will be implemented.

Table 3: GCAL1 and GCAL2 Results, CLIMB case 1000 km

	GCAL1 Value	**GCAL2 Value**	**Real Value**
CubeSat Mass	**4.03kg**	**4.03kg**	**4.2kg**
Form Factor	**3U**	**3U**	**3U**
Power Production	**25W**	**25W**	**25W**
Solar Panels Nr.	**5**	**5**	**5**
ADCS	**Closed Loop ADCS Y Mom NewSpace Systems**	**Closed Loop ADCS Y Mom NewSpace Systems**	**TBD**
Propulsion	**Enpulsion Nano**	**Enpulsion Nano**	**Enpulsion Nano**
TT&C Type	**VHF GomSpace**	**VHF GomSpace**	**S-Band In-House built**
Structure	**3U ISIS Space**	**3U ISIS Space**	**In-House built**
OBC	**ISIS OBC**	**ISIS OBC**	**TBD**
Battery Pack	**4 Li-Ion Batteries**	**4 Li-Ion Batteries**	**8 Li-Ion Batteries**

8 Conclusions

The GREATCUBE+ software tool developed by the University of Applied Sciences Wiener Neustadt in cooperation with the Technical University Dresden shall support the design process of CubeSat. The architecture and the general philosophy of the software tool has been described and the rational of it has been discussed. Using the design data of nine different CubeSats, the capabilities of GC+ have been tested. It has been demonstrated that, when iterated with realistic inputs, the current version of GREATCUBE+ is capable of providing results which are accurate at 96% at satellite level, while minor differences arise at subsystem level. The utilization of Concurrent Engineering, so far, has proven to be very effective. Both versions employed by GREATCUBE+, namely the method GCAL1 and GCAL2, are providing accurate results. In conclusion, based on the present development status, GREATCUBE+ has shown that it can effectively support the initial design phase of CubeSats. In a next step, it is planned to implement more sophisticated routines including simulations of the EPS and a numerical approach for the thermal assessment of the satellite. This will lead to an even better and more detailed design prediction of GREATCUBE+.

References

[1] JM Álvarez and E Roibás-Millán. "Agile methodologies applied to Integrated Concurrent Engineering for spacecraft design". In: *Research in Engineering Design* 32.4 (2021), pp. 431–450.

[2] *Cal Poly's Game-Changing CubeSats Inducted into the Space Technology Hall of Fame*. https://www.calpoly.edu/news/cal-polys-game-changing-cubesats-inducted-space-technology-hall-fame. Accessed: 2022-06-04.

[3] Ridolfi G. & Corpino S. & Mooij E. "A System Engineering Tool for the Design Of Satellite Subsystems". In: *AIAA* AIAA-09-6037 (2009).

[4] Sharan A. Asundi & Norman G. Fitz-Coy. "CubeSat Mission Design Based on a Systems Engineering Approach". In: *IEEEAC Paper nr 2180* (2013).

[5] Curtis PhD Howard. *Orbital Mechanics for Aerospace Engineering Students*. ISBN-100080977472: Elsevier, 2013.

[6] Dominik Knoll, Clement Fortin, and Alessandro Golkar. "A process model for concurrent conceptual design of space systems". In: *Systems Engineering* (2021).

[7] *NANOSAT Database*. https://www.nanosats.eu/. Accessed: 2022-06-04.

[8] Goran D Putnik and Zlata Putnik. "Defining sequential engineering (SeqE), simultaneous engineering (SE), concurrent engineering (CE) and collaborative engineering (ColE): on similarities and differences". In: *Procedia CIRP* 84 (2019), pp. 68–75.

[9] Chang Y. & Hwang K. & Kang S. "SEDT (System Engineering Design Tool) development and its application to small satellite conceptual design". In: *Acta Astronautica* 61 (2007) 676-690 (2007).

[10] Pablo Salgado Sanchez et al. "The ESA Concurrent Engineering Challenge in Aerospace Engineering Education: Educational Impact and Perspectives at the Technical University of Madrid". In: Oct. 2020.

[11] Catherine Venturini et al. "Improving mission success of CubeSats". In: (2018).

[12] Thyrso Villela et al. "Towards the thousandth CubeSat: A statistical overview". In: *International Journal of Aerospace Engineering* 2019 (2019).

[13] Wiley J. Wertz James R. & Larson. *Space Mission Analysis And Design (SMAD)*. New York: Springer, 1999.