

A Generic Simulink Model Template for Simulation of Small Satellites

**Axel Berres⁽¹⁾, Marco Berlin⁽¹⁾, Andreas Kotz⁽²⁾, Holger Schumann⁽³⁾,
Thomas Terzibaschian⁽²⁾, Andreas Gerndt⁽³⁾**

⁽¹⁾ German Aerospace Center (DLR), Simulation and Software Technology,
Rutherfordstr. 2, 12489 Berlin, Germany

⁽²⁾ German Aerospace Center (DLR), Institute of Robotics and Mechatronics,
Rutherfordstr. 2, 12489 Berlin, Germany

⁽³⁾ German Aerospace Center (DLR), Simulation and Software Technology,
Lilienthalplatz 7, 38108 Braunschweig, Germany

ABSTRACT

This paper presents a template architecture for a straightforward specification of small satellite missions by means of a domain specific language. Furthermore, the process to transform this model to a platform-dependent, executable simulation is depicted. As a first prototype environment, Simulink has been selected. The design adaptation during the developing process is illustrated using the power system of the OOV-TET satellite.

1. INTRODUCTION

Assembling real satellite prototypes is time-consuming and expensive. To avoid high evaluation costs which come along with such kind of design approach, the functional design of a satellite and the interdependencies of its subsystems may also be simulated. This allows the assessment of many design variations to obtain the optimum solution for a given mission scenario. Moreover, simulations are also applied in education to impart the knowledge about the functionality of satellite systems.

In the field of small satellite development, however, simulations are seldom used. The main reason for that is that each small satellite comes with its own unique design. Creating a simulation framework which merely fits to this special architecture is in general not feasible. But in recent years, a number of small satellites series has been developed. As reducing costs becomes the essential issue to be successful on this growing market, the usage of simulation approaches even in this field becomes more and more important. Nowadays, many companies and research institutes are working on generic frameworks which support the reuse of already developed simulation modules. Engineers do not have to re-develop the core simulations from scratch again and again but can now concentrate on the actual mission objectives.

In this paper, we describe a domain specific template for the Simulink simulation environment [1], which can be used to design an arbitrary series of small modular satellites. Only the content of the components of the template has to be adapted to the mission in mind.

2. SMALL SATELLITE SERIES

The basis for the development of such a universal Simulink template is the definition of a common satellite architecture for small satellite series. Therefore, in a first step, we analyzed a couple of typical examples of small satellites which are classified by their mass as pico ($\leq 1\text{kg}$), nano ($\leq 10\text{kg}$), or micro ($\leq 100\text{kg}$) satellites.

The amateur satellite czCube [2] served as an example for the first class and complies with the CubeSat specification [3]. The main objective of this project is to verify the components needed for such tiny satellites.

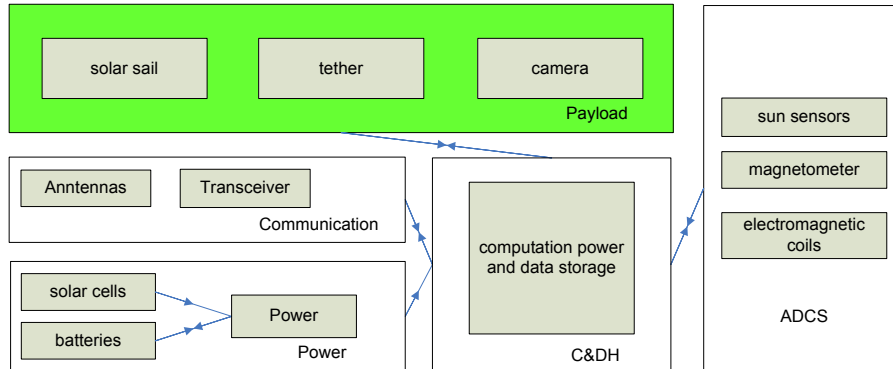


Figure 1 czCube architecture [2]

The czCube architecture shown in Figure 1 consists of an Attitude Determination Control System (ADCS,) a Communication, a Power, and a Command and Data Handling (C&DH) system. The ADCS system inherits sun sensors, magnetometers, and electromagnetic coils to determine and control the satellite attitude. To communicate with ground stations the czCube has two antennas and a transceiver as Communication System. Necessary power is generated via solar cells, stored in batteries, and distributed to the consumers by the power subsystem. All satellite components generate housekeeping data and measured values, which will be stored and send to the ground stations by the Command and Data Handling system.

However, the purpose of the czCube is to verify components, which therefore serve as payload. The Cubesat verifies a solar sail and a tether. For this verification a camera is on board to take pictures of the deployment of tether and solar sail.

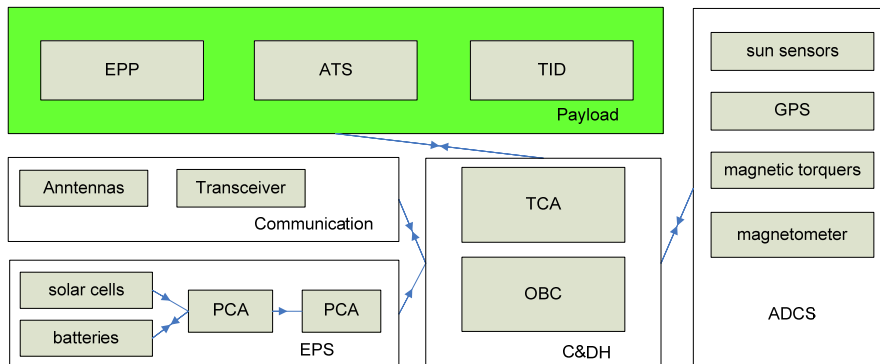


Figure 2 HAUSAT-2 architecture [4]

A nano satellite is represented by HAUSAT-2, a Korean education satellite built by SSRL (Space System Research Laboratory) of the Hankuk Aviation University. Its

mass is 25 kg. Similar to czCube, the Electrical Power System (EPS) shown in Figure 2 uses solar cells and batteries for power supply. The command and data handling (C&DH) subsystem offers the Telemetry & Command Assembly (TCA) and an On-Board Computer (OBC) needed to perform tele-commands and on orbit data handling. The Communication system consists of a transceiver and antennas. A magnetometer, a pitch momentum wheel, sun sensors, a GPS module, and magnetic torquers establish the attitude determination control system (ADCS), which stabilizes the satellite in three axes. The objectives of this satellite “... are to study the scope of activities and ecology of animals using Animal Tracking System (ATS) and collect space environment data of mission orbit from Electric Plasma Probe (EPP) ...” [4]. These two systems and a Total Ionizing Dose (TIF) meter are declared as payload.

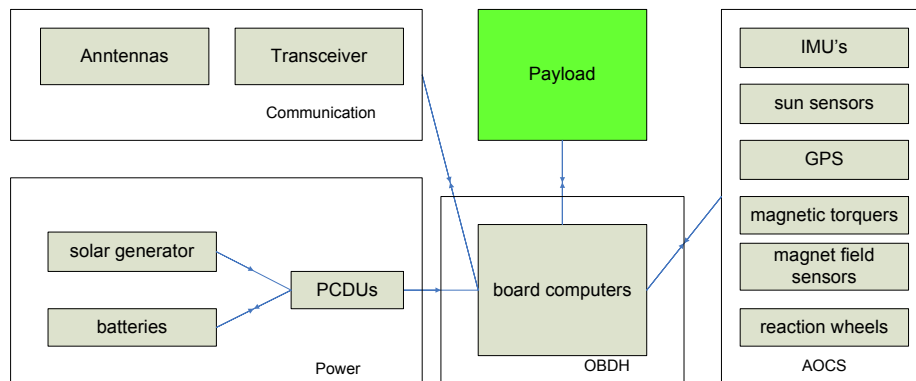


Figure 3 OVV-TET architecture

As a micro satellite proxy, we evaluated the 120 kg OOV-TET platform. The goal of this project is to design a platform for verification flights for the evaluation of newly developed techniques and technologies for space missions. All essential system components which could be identified for the operation of the other small satellite families can also be found here and are shown in Figure 3. These are for instance a power, an OBDH, an AOCS, and a communication system. As the payload is an arbitrary technology which is supposed to be evaluated, only interfaces to the satellite bus are specified.

3. A SIMULATION TEMPLATE ARCHITECTURE

Based on the experiences with the described satellite systems, we have developed a simulation template architecture suitable for all kinds of small satellite missions, which is shown in Figure 4. The design concept is based on a layer pattern. The satellite system and the environment are modelled on independent layers. The motion of the satellite has been separated from the satellite itself and is designed as a relation between satellite and physical environment in the environment layer.

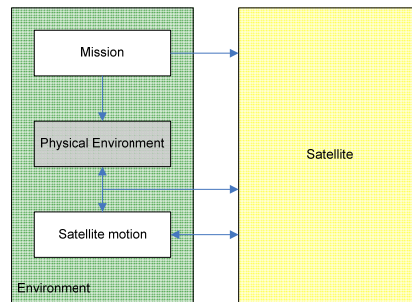


Figure 4 General satellite architecture

The environment layer contains, beside the motion of satellite, the functionality of the mission and the physical environment. Output parameters of the mission are e.g. the orbit and the mission time. Time system conversions or coordinate transformations are some of the helper functions available. During the simulation, the orbit is computed by the satellite motion. To facilitate the implementation of those methods, external tools like the satellite toolkit (STK) may be incorporated. The physical environment includes radiation fluxes from Sun, Earth or other space objects. Additional aspects like gravitation and atmosphere drag may be implemented according to simulation models defined for the ESA space environment [5].

The architecture of all previously shown satellites come along with a power, a communication, an attitude determination (orbit) control, an onboard data handling system, and the mechanical structure. All these identified subsystems are composed on the satellite layer shown in Figure 5. The core component is the structure object which describes the mechanical composition and supplies other systems with mandatory parameters. The power object offers the functionality needed to specify the power system for the simulation. The attitude determination control system is extended to an attitude and orbit control system (AOCS), which includes also orbit maneuvers. Finally, the telemetry telecommand (TTC) and OBDH provide the functionality for communication and computation. The payload, however, varies from mission to mission. Therefore, a specific mission payload is represented by a payload system template.

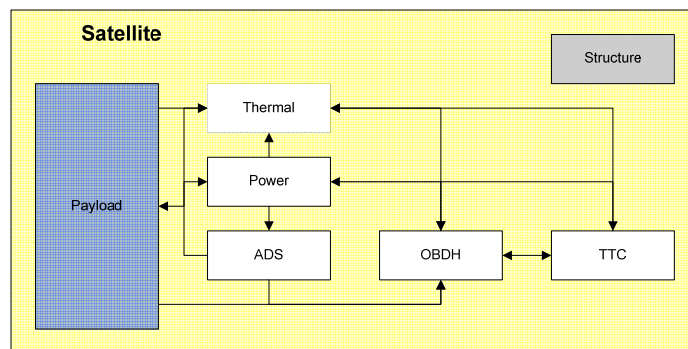


Figure 5 satellite architecture

To prove the concept of the general satellite architecture of Figure 5, an OOV-TET phase A satellite model was used. The Simulink implementation of OOV-TET is shown in Figure 6. On the left side the mission block provides the orbit for the satellite and the mission time for the environment. The environment itself calculates the position of the sun and the satellite, the fluxes coming from the Sun and the Earth, and the orbital

angular velocity of the satellite. The environmental influence on the satellite motion is not covered during a phase A simulation. In later phases dynamic perturbations due to environmental factors like the Earth's atmosphere and the solar pressure are considered.

The satellite motion implementation encapsulates dynamic behavior and kinematics of the satellite. The calculated attitude and the angular velocity of the satellite will be forwarded to the satellite.

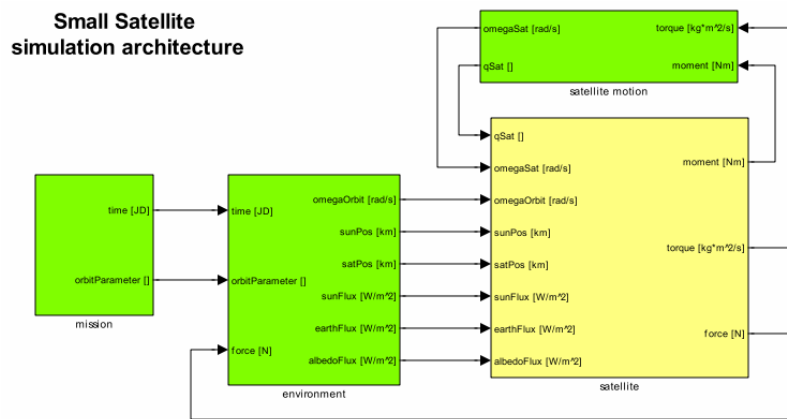


Figure 6 Small Satellite Architecture implemented in Simulink

In **Figure 7** the implemented satellite architecture is shown. Parts of the implementation are the aocs, thermal, power and payload implementation. The white marked OBDH and TTC blocks are placeholder for further simulation phases. The interface of aocs covers all possible operation modes not only to change the attitude also the change of orbits. The applied example considered three different modes leap, earth and sun pointing yet. The interfaces of the thermal and the power subsystems permit the modeling of hierarchical models.

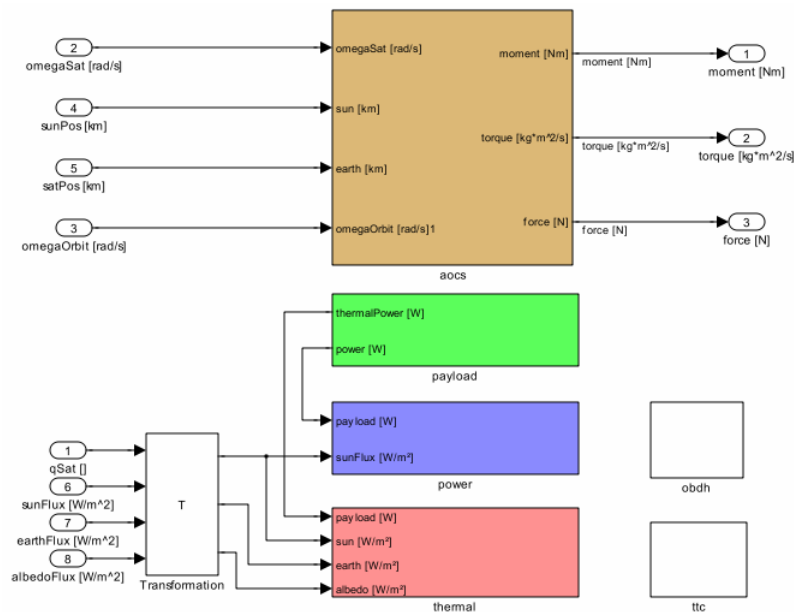


Figure 7 Implemented Satellite Architecture

4. EXECUTABLE SIMULATION GENERATION

A preferred way to specify a simulation is the model driven architecture (MDA) approach. The engineer uses a graphical editor to design all components of the model, like done exemplarily in the power system architecture of Figure 8 modeled in Simulink. At the end, the executable simulation is generated automatically. One of such general MDA tools is the Open Architecture Ware (oAW) [6], which supports all required transformation steps. For the modeling of satellite system architectures, generally SysML [7] is used. Some framework examples based on this description language is the generic satellite model [8] and the virtual spacecraft design [9].

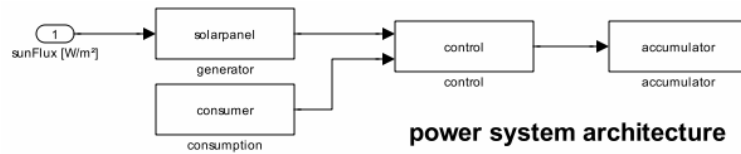


Figure 8 Sample Power System Architecture

However, a SysML description can not be used to compile an executable simulation directly but generates source code mainly in C++. Then, this code must be edited to add required behavior to the model. A disadvantage of this method is that a space engineer needs enhanced programming skills. To simplify this task, we have developed a domain specific language (DSL) to describe satellite architecture on a higher level. It enables an engineer to characterize specific satellites by changing and configuring subsystems. The error prone and time consuming implementation of simulation models coded in C++ can be avoided.

After the user has defined the satellite architecture by means of our DSL, the result can be transformed to a platform-specific model. For this step, based on oAW, we have developed a generator for Simulink which is used as the simulation runtime environment. The advantage of this way is that Simulink is well known and popular in the space engineering domain. As an engineer is already familiar with the developing of simulation models in Simulink, it is convenient to use this environment for the remaining step to create an executable simulation. However, it is not mandatory for the user to model own Simulink blocks. For the template satellite architecture, all components have been modeled already in advance and are stored in a library. The DSL-to-Simulink generator now selects appropriate blocks from the library. To depict the transformation process in more detail, the power system of OVV-TET is chosen as an example (

Figure 9). The design consists of a solar panel, a power control unit, an accumulator, and a consumer. Each component including its interfaces can be specified by means of the DSL.

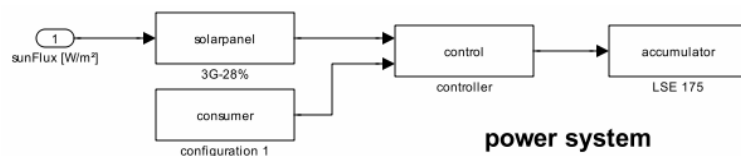


Figure 9 Power system of OOV-TET

```

System OVV-TET {
  ...
  power{
    input sunFlux;

    generator from 3G-28% {
      efficiency = 28 [%];
      area = 1 [m^2];
    }

    consumer from "configuration 1" {
      consumption = 75 [W];
    }

    control from controller {
    }

    accumulator from "LSE 175" {
      capacity = 183 [Ah];
      voltage = 3.7 [V];
      state = 95 [%];
    }
  }
  ...
}

```

In order to create a model of a specific device, a selected component has to be configured by setting component parameters. The component interface, however, never changes which emphasize the template character of our approach. After the system description is performed, the transformation of the simulation follows. A generator translates the textual description into a Matlab script. For each component, the connection among components and component parameter are translated into Matlab commands to create a Simulink model. Executing this script builds the power system with the chosen components and the specified parameter configurations. The internal check mechanism of the DSL editor prevents inconsistencies of the described system and guarantees an executable simulation.

In Figure 10 the simulation results of the power model with two different accumulator types are shown. The left picture shows the charge state of 183 Ah Li-Ion and the right picture shows the charge status of a 100 Ah Ni-H₂ accumulator. The satellite starts in the Earth shadow, until it comes after roughly 1800 seconds into the sun and the accumulator could be charged. The full capacity of the Li-Ion accumulator is reached after roughly 4000 seconds, whereas the Ni-H₂ accumulator needs 4300 seconds to be fully charged. At 4500 seconds the next eclipse starts.

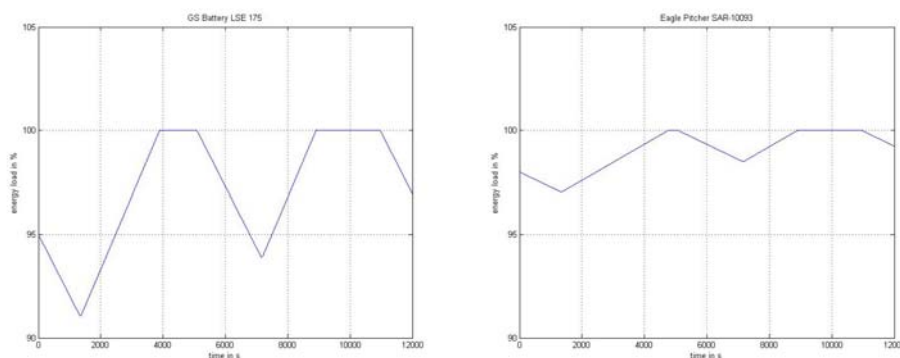


Figure 10 OOV-TET accumulator states of different accumulator types

5. CONCLUSION

We have presented a template architecture for the specification of small satellite missions that simplifies the way from a first design to an executable simulation. We selected Simulink as simulation environment in order to evaluate our approach. On the basis of OOV-Tet's power system, we could prove that we offer a seamless framework for a straightforward, domain specific component modeling. In general, other environments can be used as well. The DSL would be the same but the platform specific transformation process must offer the required proprietary simulation blocks. To

support heterogeneous simulation frameworks, the Simulation Model Portability Version 2 (SMP2) standard has been developed [10]. Right now, this is the most important simulation exchange format used in European aerospace projects. Therefore, we are going to focus on the transformation process for such environments.

6. REFERENCES

- [1] H. Klee, *Simulation of Dynamic Systems with MATLAB and Simulink*, CRC Press (2007)
- [2] A. Holub, *CzCube: Czech Amateur CubeSat*, Proceedings, 2. European CubeSat Workshop, ESA/ESTEC, Noordwijk, The Netherlands, January 20-22 (2009)
- [3] CubeSat Design Specification. Available online at: http://cubesat.atl.calpoly.edu/media/CDS_rev11.pdf (accessed February 2009)
- [4] Y.-K. Chang, S.-J. Kang, B.-Y. Moon, and B.-H. Lee, *Low-Cost Responsive Exploitation of Space by HAUSAT-2 Nano Satellite*, Proceedings, 4th Responsive Space Conference, AIAA, Los Angeles, CA, USA, April 24-27 (2006)
- [5] ECSS Space Environment Standard (ECSS E-10-04). Available online at: http://space-env.esa.int/ECSS/ecss_10_04.html (accessed February 2009)
- [6] openArchitectureWare User Guide, Version 4.3.1. Available online at: <http://www.openarchitectureware.org/pub/documentation/4.3.1/openArchitectureWare-4.3.1-Reference.pdf> (accessed on February 2009)
- [7] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML*, Morgan Kaufmann Publisher, (2008)
- [8] B. Kraft, *Design of a Generic Satellite Model*, Semester Thesis, TU Munich (2008)
- [9] J. Fuchs, *Der virtuelle Entwurfsprozess (Virtual Spacecraft Design –VSD)*. Available online at: http://www.dlr.de/sc/Portaldata/15/Resources/dokumente/WS_120607/Presentation_VSD_070612_Fuchs.pdf (accessed February 2009)
- [10] P. Fritzen, *Model Re-use through the SMP 2 Standard*, Proceedings, RAeS Simulation of On-board Systems, The Royal Aeronautical Society, London, November 3-4 (2004)