Open Source CubeSat Workshop 2018

Monday 24 September 2018 - Tuesday 25 September 2018

European Space Astronomy Center (ESAC/ESA)

Book of Abstracts

Contents

Welcome and Keynote / 50

Welcome from Organizing Team

Author: Artur Scholz[1]

[1] *LibreCube*

Corresponding Author: artur.scholz@librecube.net

Welcome and Keynote / 51

Welcome from ESAC

Author: Danny Lennon[1]

[1] *Head of Science Operations Division, ESAC*

Welcome and Keynote / 52

Welcome from Libre Space Foundation

Author: Manthos Papamtthaiou[1]

[1] *Libre Space Foundation*

Welcome and Keynote / 53

Keynote: Open Source Isn't Rocket Science, but Rocket Science is Open Source!

Author: Bruce Perens[1]

[1] *Open Research Institute, Inc.*

Corresponding Author: bruce@perens.com

Talks / 12

Formation Flying, an opportunity to enhance microwave cosmology with CubeSats

Authors: Javier Cubas[1]; Francisco Javier Casas Reinares[2]; Enrique Martínez González[2]; Juan Bermejo Ballesteros[1]; Ángel Sanz Andrés[1]

[1] *Instituto Universitario de Microgravedad Ignacio Da Riva, Universidad Politécnica de Madrid*

[2] *CSIC*

Corresponding Author: j.cubas@upm.es

CubeSat has become a satellite concept with increasing interest for space research and exploration. Key elements of its success have been standardization, easy design and low cost. However, the constraints of space, weight and power represents practical limitations for the scientific instrumentation that can be accommodated limiting the scientific scope that the CubeSat concept can have. Nevertheless, it is expected to be a very competitive solution for those scientific applications that can be adapted to those constraints.

Recent missions such as CanX-4 & 5 have shown that CubeSats already have the necessary technology to carry out formation flight. Formation flying multiplies the possibilities of performing science of small satellites. This concept is the one used in the future ESA PROBA-3 mission, in which one of the satellites of the formation flight acts as a parasol, hiding the solar disk so that the other can study the solar corona. Although PROBA-3 is not made up of small spacecrafts, other missions with similar concept, like NASA's CANYVAL-X, are based on CubeSat technology to demonstrate how two small satellites in formation can be used as a single large telescope. Finally it is possible to use small satellites in conjunction with a large scientific instrument. The CubeSat would travel as a piggyback next to the main ship and could be deployed once in orbit. A small satellite deployed in this way could be used, for example, as a calibrator for the instruments of the main ship, thus obtaining more precise measurements. Furthermore, the small weight and cost of the CubeSats would allow them to be added to the main mission without having a great impact, and they can be conceived as science enhancements of the main mission without jeopardizing it.
In this talk, we present the possibilities of CubeSat formation flying for scientific research in the context of the Cosmic Microwave Background (CMB) science. We will show how the CubeSat concept can be of interest to accommodate a calibration system in the microwave range that can be used to calibrate CMB instrumentation, based either on the ground and using a LEO for the calibration satellite, or on a space mission operating at L2 and requiring formation flying for the calibration satellite. The possibility of having a calibration satellite for CMB science will provide an unprecedented control of systematic effects that are the limiting factor of the future very sensitive CMB experiments.

Talks / **30**

Millimetre-wave polarisation calibration CubeSat

Authors: Mansur Tisaev[1]; Marcos López-Caniego[2]; Julio Gallegos Alvarado[2]; Xavier Dupac[2]; Jorge Fauste[2]; Fernando Martin Porqueras[2]

[1] *ESA/ESAC; University of Bristol*

[2] *ESA/ESAC*

Corresponding Author: mansurtisaev@gmail.com

Investigating the polarisation of the cosmic microwave background (CMB) provides us with the opportunity to study the formation of the early universe, in particular it reveals the effect of gravitational waves caused by inflation after the Big Bang. Experiments studying the CMB polarisation use ground and balloon-based polarimeters operating in the millimetre wave range, and these require both precise and absolute measurement of the polarisation orientation. Ideally, the orientation will be measured to within 0.01 degrees, which will allow systematic errors of the detector and false polarisation signals to be filtered out. The required precision can be achieved with the use of a known-source acting as a calibrator, for which a CubeSat is proposed. The satellite contains several linearly polarised sources in the sub-100GHz frequency range, with the exact polarisation orientation determined through pre-launch testing, as well as an accurate attitude determination and control system (ADCS). It could serve as a calibrator for current CMB polarimeters, such as QUIJOTE in the Canary Islands and ACTPOL in Chile, as well as for future European polarisation studies.

Work on this CubeSat project focused on selecting the signal frequencies required by the detectors and specifying the components needed to achieve optimal signal-to-noise ratio at the polarimeters. This led to the design of a calibration CubeSat, which includes the layout, orbit, power budget and testing requirements, with prototypes manufactured for layout definition and to test the concept

using a drone. Future work will focus on an integrated CubeSat design with a precise and agile ADCS, as well as more refined predictions of the satellite's operation and possible further testing on a high altitude balloon.

Talks / **36**

CanSat: The best way to start getting involved in space

Author: Daniel Sors Raurell[1]

Co-author: Mark Cowan [1]

[1] *Open Cosmos*

Corresponding Author: dani@open-cosmos.com

The idea of simplifying space missions by using CubeSats and CanSats appeared around 20 years ago with the main purpose of making space technologies accessible to a broad audience. Both platforms proved to be a game changer in the space sector up to the point nowadays that anyone can start developing space related missions from home.

The CanSat platform has been used as the perfect cultivation pot to promote STEM fields among the young generations and inspire the soon to become engineers and scientists. It is the perfect introduction to space science and technologies while usually being the first experience into a simulation of a real space mission.

An Open Source development platform at those first stages has proven to be of high impact and interest not only for Secondary and High Schools students but also at University level. Open Cosmos developed qbcan, a CanSat development kit composed of affordable COTS with the idea of simplifying the process of getting people involved into software and electronics using space as a context. A set of educational wiki pages were created with libraries and sample codes so that even non-technical and non-experienced users could easily assemble a CanSat and conduct a primary mission. A forum was also created and maintained to provide support to the increasing CanSat community with all the designs and list of components available following the Open Source philosophy.

The presentation will review the rationale behind the qbcan open source platform, highlighting the CanSat concept and its contribution in making space more accessible. A short demonstration and hands-on workshop could be also considered for the audience to understand much better the platform and components used.

Talks / **38**

Enabling Open and Remote-Access Sensing on Small Satellites

Authors: Christian Gomez[1]; David Salcedo[1]; Emmanuel Gomez[1]; Fernando Vega[1]; Juan Vargas[1]; Sebastian Agüero[1]; Gustavo Santos[1]; Juan A. Fraire[1]; Pablo A. Ferreyra[1]; Marco A. Reyna[1]

[1] *Comisión Nacional de Actividades Espaciales (CONAE)*

Corresponding Authors: juanfraire@gmail.com, malvarezreyna@vengsa.com.ar

It is not new that satellite design, component procuration, integration, testing, launch and operation are cost and time demanding. However, during the last decade, the small-satellite concept re-emerged not only as an enabler of smaller and lightweight platforms, but also as a new aerospace paradigm based in a) the adoption of up-to-date consumer technologies, b) rapid development cycles

and c) small agile teams operating closer to IT industry management models rather than those found in traditional military/aerospace organizations.

In this context, we further push the paradigm to propose Open and Remote-Access Sensing (ORAS) as a mean to share the utilization of in-orbit resources of a small cubesat among several users with autonomous and potentially isolated ground stations. The ORAS concept and the specifics of a mission currently under design to implement it are product of a joint effort of professor and students of master on Satellite Instrumentation course held at the Gullich institute, within the Argentinian Space Agency (CONAE) in Córdoba, Argentina.

The main driver behind the ORAS project is the fact that state-of-the-art technology for modern small platforms tend to provide large storage and processing capabilities, but limited downlink datarate, typically constrained by available in-orbit power and limited antenna gains. Indeed, both the placement and accurate pointing of a high-gain antenna as well as energy harvesting via solar panels within the size of a cubesat are difficult endeavours. As a result, the downlink communication channel quickly becomes a bottleneck hindering the efficient use of the platform.

To balance and improve the utilization of valuable in-orbit resources we propose ORAS: an open-access strategy where satellite's instruments are accessible to as many users as possible. This allow any ground station equipped with a standard communication hardware publicly available in the market to access, command, manage and download instrument acquisitions. In particular, an acquisition might be executed and previewed via thumbnails by users but not necessarily downloaded. As a consequence, the downlink channel utilization is better utilized and extended for as many users interested in the observations thus widening the resource bottleneck and maximizing the scientific return of the orbiting object.

Nonetheless, challenges to successfully achieve ORAS are not minor. Allowing anyone anywhere to access the satellite requires of non-trivial but highly developed protocols that a) provides secure access to authorized and authenticated users to command specific and carefully chosen functions of the platform and b) ensure a correct utilization of observation resources by keeping and administering an in-orbit schedule and avoiding battery-power exhaustion. Since the mission control cannot be continuously in contact with the cubesat, all these functionalities need to be performed autonomously in the cubesat.

Besides describing in details how these unprecedented challenges can be tackled, this presentation discusses how a specific cubesat mission can implement the ORAS concept by making use of Components Off The Shelf (COTS) in redundancy and leveraging fault-tolerant architectures to increase the reliability of the system to meet the expected lifetime of the mission. We analyze the case of an instrument devoted to observe termic-sensitive events on the surface, a thermal camera which allows to monitor thermal anomalies and warn of possible emergency situations. In particular, we discuss the monitoring of volcanic activity on the Andes mountain as a motivational example.

Talks / 31

Open source hardware and software PocketQube family of satellite modules

Authors: Ilias Daradimos[1]; Pierros Papadeas[1]

[1] *Libre Space Foundation*

Corresponding Author: ilias@libre.space

Libre Space Foundation is designing an open-source hardware and software family of PocketQube format satellite modules. We are focusing on the main module including typical OBC and COMMS functionality delivered using modern and high capable MCUs (STM32) and reconfigurable COMMS ICs (AX5043). Data logging and power management functions are also present. We are following the mechanical specifications of the proposed PQ9 standard while adapting the electrical and application layers to meet modularity and compatibility needs. Several design aspects will be addressed including subsystem and software design for

reusability and alternative inter-subsystem communication methods to the proposed RS485 communication of the PQ9 standard.

Talks / **8**

The LibreCube Ecosystem: How to Use it, How to Contribute

Author: Artur Scholz[1]

[1] *LibreCube*

Corresponding Author: artur.scholz@librecube.net

The LibreCube Initiative has the objective to promote and develop open source projects that form modular and space-ready elements, which then can be used for earth and space exploration missions, mainly CubeSat missions. To achieve this, LibreCube is based on three pillars: open source, standardization, and a reference architecture.

While the open source aspect of LibreCube is a very important ingredient (and applies likewise to the outputs but also to all the tools used for producing the outputs), the other two aspects are of equal importance to enable true collaboration and sharing among the community.

Standards (in particular interface standards) are essential to ensure that LibreCube elements are compatible to each other. It is not that the engineering world is short of standards. However, usually standards are hard to read and it is not easy to decide which standard to apply. LibreCube has defined a clear policy on what standards to choose. They must be openly available (downloadable from internet), free of costs, and preferably based on space heritage. The two major organizations that publish such standards are the European Cooperation for Space Standardization (ECSS) and the Consultative Committee for Space Data Systems (CCSDS).

While standardization and open source path the way for collaboration, it is the definition of a common space system architecture that allows contributors to work on different parts of the system in parallel - ensuring that eventually it will fit into a large system. The generic reference architecture provides the big picture for this and ties all the efforts together.

This presentation will thus give an overview picture on the standardized interfaces and data exchange protocols for space and ground elements, as defined by LibreCube. It will then outline the development cycle of prototypes and elements, to show how projects materialize from idea to operational product. For the potential users of LibreCube elements it will be presented on how to reproduce (assemble) and operate them. Finally, for potential developers it will be explained how to contribute to new or ongoing prototypes and elements.

Talks / **40**

SatNOGS - Open source ground station network

Author: Fredy Damkalis[1]

[1] *Libre Space Foundation*

Corresponding Author: pierros@papadeas.gr

An overview of the SatNOGS project, a network of satellite ground stations around the world, optimized for modularity, built from readily available and affordable tools and resources.

The trend of Low Earth Orbit (LEO) satellite launches continues and new countries and players join the space journey. SatNOGS provides a scalable and modular platform to track, receive telemetry, monitor and command & control these satellites. SatNOGS technology stack including hardware ground station designs (antennas, rotators, electronics), software for SDR-based communications, scheduling and observations, are developed as open source projects with hunderds on contributors from around the world.

Current operations support VHF and UHF bands, while also experimenting with higher bands like L, S and X. Furthermore SatNOGS provide an easy way to store, access and view increasingly received satellites data. (db.satnogs.org, more than 17 million packets so far)

SatNOGS continues to grow, develop and improve as an infrastructure allowing observers to take advantage of this networked ground segment and remotely operating SatNOGS ground stations around the world.

Talks / 9

Developement of a CubeSat communications system based on CCSDS and ECSS standards

Author: Milenko Starcik[1]

[1] *TU Darmstadt Space Technology e.V.*

Corresponding Author: starcikm@gmail.com

TU Darmstadt Space Technology e. V. is a group of students at the *Technical University of Darmstadt*. Its purpose is to get students in contact with space technology. To give students the opportunity to get first hands on experience in practical projects, there are currently a CubeSat and experimental rockets in development. Because also others should benefit from this work, all the CubeSat Hardware and Software are published as open source. The licenses used are CERN OHL v1.2 for Hardware and GPLv3 for Software.
The mission of the CubeSat is to test a transparent reflectarray, a combination of a solar cell and a reflectarray antenna. The mission is being developed in cooperation with the *Institute of Microwave Engineering and Photonics* at the *Technical University of Darmstadt*.
The focus, when developing a communications system for CubeSats, should always be on the reliability of the overall system, to be always in control of the satellite. To ensure this, the *Consultative Committee for Space Data Systems* (CCSDS) and the *European Cooperation for Space Standardization* (ECSS) developed standards for reliable communication systems. Using these standards, it is possible to realize all kind of mission concepts and control multiple satellites with one or more groundstations. The ECSS standards for communication are derived from the CCSDS standards, with a focus on user friendliness and they are in most parts compatible with each other.
To get fast results in the development process, software defined radios (SDR) are used with GNUradio and Python scripts for data processing. The printed circuit boards (PCB) are developed using the open source PCB design software KiCAD and build with commercial of the shelf parts, to keep the development and production costs low.
In a first step of the development the low layer data transmission is being implemented in GNUradio. A demonstration was presented at the *Open Source CubeSat Workshop 2017*, with packet detection inside of GNUradio. The overhead and the complexity of the project made it hard to continue with the development, so the processing inside of GNUradio is now just done until the step where the raw bit stream is created. After this is done the bits are passed via zero message queue protocol (ZeroMQ) to Python scripts that implement the standards for Synchronization and Channel Coding (CCSDS 131.0-B-3, 231.0-B-3), Communications Operation Procedure-1 (CCSDS 232.1-B-2) and the still not finally reviewed Unified Space Data Link protocol (CCSDS 732.1-R-3).
For the operation of the CubeSat the possibilities offered by the Message Abstraction Layer (CCSDS 521.0-B-2) and the Mission Operations Services concept (CCSDS 520.0-G-3) are currently being reviewed.

Talks / 5

ESAT, the educational satellite

Author: Ignacio Barrios Tascon[1]

Co-author: Jose Miguel Ezquerro Navarro [1]

[1] *Universidad Politécnica de Madrid*

Corresponding Author: jezquerro@eusoc.upm.es

The Universidad Politécnica de Madrid (UPM) started in the academic year 2009/10 educational innovation activities in space engineering. The implementation of several topics in the Conceive - Design - Implement – Operate (CDIO) syllabus was pursued, mainly focused on the next topics: Hardware and software integration, Test, Verification, Validation and Certification.
With this objective the use of demonstrator satellites was included in the practical lessons. The development of a self-designed demonstrator satellite was started leading to the ESAT's birth.
ESAT has been created and developed by Theia Space, an initiative born at the Spanish User Support and Operations Center (E-USOC) which belongs to the UPM and is one of ESA's delegated centers for the operation of scientific payloads onboard the International Space Station.
ESAT is an educational satellite designed for hands-on learning for all education levels: STEM education, university studies and professional training. It is a 10x10x10 cm nanosatellite based on the successful CubeSat standard and weighing less than 1 kg.
ESAT has the typical spacecraft subsystems: Electrical Power, Command and Data Handling, Attitude Determination and Control and Structure.
The user can choose to focus and work on each subsystem independently or to practice with the fully integrated satellite. ESAT features a Wi-Fi communication system allowing the connection with a PC, where the ESAT GUI based on COSMOS SW allows an easy operation of the satellite.
ESAT is perfectly fitted to train on design, manufacture, integration, validation and operation of satellites.
It has been developed with the open source philosophy and the users are able to expand its functionalities. ESAT allows to integrate and test new user developments, both SW and HW. For the HW developments the user has access to all the lines in the satellite, including power and communication lines and analog and GPIO lines.

Talks / **45**

MAGIC: a miniaturised magnetometer for space weather monitoring with CubeSats

Author: Chiara Palla[1]

Co-authors: Patrick Brown [1]; Henry Eshbaugh [1]; Tim Oddy [1]; Jonathan P. Eastwood [1]

[1] *Imperial College*

Corresponding Author: c.palla@imperial.ac.uk

In situ space weather measurements are currently sparse; this represents an obstacle to further improving our understanding of the Sun-Earth interaction. Constellation missions are ideal as they enable a comprehensive and broader set of magnetic field data at many points simultaneously. A novel approach is to exploit CubeSat platforms, however, on them, mass and power resources are limited.
MAGIC (MAGnetometer from Imperial College) is a miniaturised magnetometer, which optimises noise performance while minimising power consumption by utilising a hybrid anisotropic magnetoresistive (AMR) sensor triad. This solution is suitable for CubeSats: it has flown on three of them to date (TRIO-CINEMA) and an improved design was developed for the Sunjammer microsatellite. On CINEMA, the instrument was able to detect magnetic field fluctuations associated with field aligned currents over the northern auroral oval.
An evolved version of MAGIC instrument will fly on board RadCube as part of the RadMag payload. RadCube is a 3U CubeSat funded by ESA under the IOD (In-Orbit Demonstrator) GSTP, scheduled for launch in 2020. The aim of the mission is to demonstrate miniaturised instrument technologies

in LEO for space weather monitoring purposes.
The primary scientific goal of MAGIC is to improve the understanding of field aligned currents and ring current during geomagnetically disturbed conditions for space weather monitoring.
Consequently, the opportunity to openly share MAGIC data with the scientific community, could improve our capabilities to predict space weather phenomena such as geomagnetic storm, but also understanding of substorms (in terms of occurrence, severity, duration).
MAGIC on RadCube will include 2 AMR three axes DC sensors: one in-board, and one out-board, housed in a mechanical chassis, deployed by means of tape spring motorized boom.
Although the main sensor and control loop is at TRL 9, the proposed magnetometer design on RadCube includes some technical development. For example, a major addition from the CINEMA design is the inclusion of intelligence via the addition of an Atmel microcontroller. This enables use of standard communications protocol to the bus. Furthermore the instrument will be implemented on a PCB compliant with CubeSat form factor, with components optimised for a longer lifetime mission than CINEMA. The target sensitivity is less than 2 nT.
Achieving this improved design will enable an optimised and more resilient magnetometer instrument, implementable as a "plug and play" sensor on CubeSat platforms, to be used either in a constellation configuration or as single hosted payload, for space weather monitoring in the context of ESA Space Situational Awareness's D3S monitoring concept.

Talks / 44

Ecuadorian Synchrotron Space Probe "BUHOSAT"

Authors: Daniel Vera[1]; Ericson Lopez[2]

[1] *Quito Astronomical Observatory*

[2] *Quito Astronomical Observatorio*

Corresponding Author: ericsonl@hotmail.com

The "BUHOSAT", which adopts its name from the university mascot, is a unique 1U CubSat with an astronomical mission to detect synchrotron radiation. The team composed of scientists and engineers designed a 1U CubeSat capable of carrying this novel mission. The BUHOSAT is the perfect example of the use of current enabling technologies for nano-satellites, commercial of the shelf (COTS) plus indigenous patented technology. It carries two payloads, one that will be used to study the universe and the other one for Earth observation; has quick mobility over one axis to maintain desired attitude during its orbit; provides enough power to carry out its rigorous mission; has 3 independent processing units, and uses active and passive thermal controls; all in one small package of 1U and under 100,000 USD. Most of the design, in this project, can be considered finished. Minimal changes will be done after simulation and testing is completed.

Design, implement and put into orbit a nano-satellite that allows us to obtain the basic scientific and technological experience to continue with more complex experiments and projects within the Ecuadorian Satellite System. Whose potential applications, depending on the need, will be: the radio link of automatic stations for measurement meteorological parameters, photographs of the surface of the earth, in the visible and in infrared parts of the electromagnetic spectrum, to monitoring: temperature changes in volcanic zones, country borders and natural resources.

Talks / 37

Live video from space using COTS hardware and open source software

Author: Alexandru Csete[1]

[1] *Copenhagen Suborbitals*

Corresponding Author: oz9aec@gmail.com

Copenhagen Suborbitals is a crowd-funded, non-profit, amateur space programme with the goal of sending one of us into space, on a sub-orbital spaceflight, using our home built space capsule and rocket. Since the beginning of 2008, we have flown five rockets, two mock-up space capsules, and reached significant milestones like building bi-liquid rocket engines, active guidance and establishing a framework for regular sea-launches.

An important part of successful crowdfunding is continuous outreach and contact with our supporters. To that end, we are providing live webcasts from all major test and launch campaigns. Beginning with our Nexø-class rockets, the webcasts also include live, high definition video transmitted from the rocket.

In this presentation, I will describe some of the DIY concepts we are applying throughout the project, with focus on building a scalable video downlink and live streaming solution using COTS hardware and open source software.

Talks / **41**

Role of Open Source in the Development of a Student-Built Nanosatellite and its Team Members

Authors: Tanuj Kumar[1]; Shubham Sharma[2]; Saurabh Raje[3]; Kushagra Aggarwal[2]; Abhishek Goel[1]; Vatsal Badami[3]; Gautam Pathak[3]; Parth Kalgaonkar[1]

[1] *Department of Electrical & Electronics Engineering, BITS, Pilani*

[2] *Department of Electronics & Instrumentation Engineering, BITS, Pilani*

[3] *Department of Computer Science & Information Systems, BITS, Pilani*

Corresponding Author: kmrtnjscnc@gmail.com

Team Anant is a research oriented project by an undergraduate team of BITS, Pilani to create India's first hyperspectral imaging nanosatellite. Started in 2013, the project is currently in the Proposal Design Review (PDR) stage being conducted by the Indian Space Research Organization (ISRO).

This paper will discuss the essential role of open source in the development of this satellite and how it lets us modify code & understand fundamentals, all within limited resource situations.

The first section of the paper will be an introduction to the concept of the entire project followed by a brief on how open source supports our research-first ideology. The use of open source certainly comes with its own challenges too, and lessons learnt from those will be discussed as well.

The second section will delve into specific details of open source works used by the team, like PetaLinux as our Operating System, Git for version control, certain free IPs for a compression algorithm on an FPGA & Arduino for hardware testing, and how they specifically serve many needs of the satellite architecture.

The third section will talk about lessons learned on, and the need for proper licensing in this field, so that credit where due is given while maintaining the spirit of openness. Advantages and disadvantages of different licenses shall be discussed.

Finally, the paper will conclude with a brief on the impact of open source on the profiles of the team members, having provided them with many sought-after skills in both academia and industry. This shall be followed by proposals to increase open source use in student nanosatellites, and how we expect to use more of it in our project.

Talks / **23**

Development, and Present Status of PocketQube 'Nepal-PQ1'

Authors: Rakesh Chandra Prajapati[1]; Saurav Paudel[1]

[1] *ORION Space*

Corresponding Author: rakeshchandra.prajapati@gmail.com

I can share my experience about starting hands-on based space education in engineering education. I have been promoting space education in Nepal, where there was not any project on space. I founded my company called ORION Space to promote space education, and initiate space related project in Nepal. I started with CanSat Project, which is Project Based Learning method to teach about space and satellite technology. CanSat is a model used to teach about satellite. Now, we have a real satellite project on going in-house (in Nepal). The satellite is 5cm x 5cm x 5cm size, and its called PocketQube. We have completed OBC, and Communication board. Therefore, I can share my experience how to start space project at minimum budget. Many developing countries can learn from our experiance. We have presented our work at OSCW-ESA (Open Souce CubeSat Workshop) 2017, The 1st IAA North East Asia Symposium on Small Satellites- Serving the Needs for the Benefits of the Region in Mongolia, and conference ISTS 2017 in Japan.

Talks / **22**

Open Source Lessons Learned with Open MCT

Author: Jay Trimble[1]

[1] *NASA*

Corresponding Author: jay.p.trimble@nasa.gov

In 2014, a software development team from NASA's Ames Research Center (ARC), working in collaboration with NASA's Advanced Multimission Operations System (AMMOS), set out to build a new software system for data visualization for missions. Our functional goal was to build a modular multi-mission software system that empowered users to build and compose their own visualizations of data across mission domains, with no programming required from the users to assemble these displays. Our broader goal was to build an open software system that enables participation across government and private space programs, as well as outside the space community. The software, Open Mission Control Technologies (Open MCT) is in use on operational CubeSat missions, including the Mars Cube One (MarCO) missions and the Arcsecond Space Telescope Enabling Research in Astrophysics (ASTERIA).

Open source enables flexible use, and reduces or eliminates the proprietary nature of software that can impede collaboration. While still in the early stages, we have built a community of users and contributors, with participation inside and outside of the space community. The model for collaboration is to empower missions by enabling them to adopt the software as their own, make modifications and contributions, and see those contributions used in a larger space community. All missions benefit from the larger user base enabled by open source, as each critical eye on the software results in improvements. The Open MCT user base ranges from missions, to industry outside the space industry, to research and student projects.

The participation of the open source community has resulted in improvements to code, usability, documentation and feature suggestions. Open MCT is available on GitHub at https://github.com/nasa/openmct.
An informational web site with tutorials, documentation and an online demo is available at https://nasa.github.io/openm

Talks / **13**

Finding an Open Least Common Denominator for Live Integration of Non-space-system-standard Components within Constraint Budgets

Author: Ruediger Gad[1]

[1] *Terma GmbH, Germany*

Corresponding Author: ruga@terma.com

The practical results presented in this talk were gained with a range of Open Source Software (OSS) solutions. The most important employed OSS is "bowerick" [1], which is a Message-oriented Middleware (MoM) wrapper around Apache ActiveMQ [2] and various supporting libraries such as Eclipse Jetty [3] and Spring Messaging/Websocket [4] for WebSocket-based [5] transport or Eclipse Paho [6] for MQTT-based [7] transport. bowerick and its dependencies are available under commercially friendly OSS licenses such as the Eclipse Public License, the Apache License, or the MIT License.

Space system standards such as PUS [8] or CCSDS-MO Services [9] aim on enabling interoperability between different systems or components. Yet, there are still situations that require the integration of components that do not support these space system standards. Furthermore, these standards are sometimes comparably heavy-weight with respect to the implementation effort such that, when no pre-existing libraries can be used, their adoption would require significant budget.

Small projects such as studies, prototypes, or small satellite operations may suffer from strongly constraint budgets. Hence, it may not be possible to implement heavy-weight standards when no pre-existing libraries are available. Our focus is explicitly on implementation effort and interoperability and not on resources like CPU, memory, or network throughput because, usually, the largest cost factors for small projects are the actual development and integration effort and not computing infrastructure related hardware resources.

In this talk, we report our experiences with other more light-weight implementation OSS alternatives for easing and reducing the cost for the live integration of non-standard components. The presented results are partly based on experiences gained during two studies regarding the integration of ESA software systems with Virtual Reality (VR) [10] and Augmented Reality (AR) devices.

For enabling live integration at run-time, we consider the aspects serialization format and communication infrastructure protocol. Popular serialization formats are, e.g., XML, Google Protocol Buffers, JSON, and to some degree CORBA. Popular communication infrastructure protocols are, e.g., TCP/IP Sockets, CORBA, or MoM-based solutions based on protocols such as STOMP, OpenWire, MQTT, or Web-sockets.

In the talk, we will first contrast key aspects of these formats and protocols. Based on a conceptual assessment and practical results, we selected the combination of JSON UTF-8 String byte arrays for serialization and the MoM protocols OpenWire [11], STOMP [12], MQTT [7], and STOMP over WebSockets [6] as communication infrastructure.

In our scenario, all protocols can be used equivalently as they are all supported by the ActiveMQ broker and the broker forwards between them. However, while the broker supports automatic exchange of messages between these protocols it showed that even simple combinations suffered from conversion problems.

As simple example we consider the exchange of String messages between OpenWire and STOMP. The broker falsely converted STOMP StringMessages to OpenWire ByteMessages because of misinterpreting a header field, while the opposite direction worked. More complex conversions pose even more potential pitfalls.

By using byte-array-based messages only, all so far encountered conversion problems could be avoided and messages can be transparently exchanged between the supported protocols. Thus, the selection of the MoM protocol only depends on the availability of libraries and implementation effort.

In addition, JSON-based serialization is well supported in many programming languages. Moreover, even if no JSON de-/serializer implementation is available it can typically be easily implemented.

In the VR- and AR-related studies, it showed that these capabilities, having a well-supported lightweight implementation serialization format and transparent support and exchange between MoM protocols, significantly eased development and integration. E.g., the development of a key component was done in Python with STOMP as protocol because of the availability of well-featured libraries. The VR application was based on the Unity 3D/VR framework and thus was implemented in C#. Because of the availability of a well-featured C# library, OpenWire was used as protocol for the VR application. Furthermore, we performed experiments with web-based VR, which used STOMP over WebSockets for connecting to our integration infrastructure.

On the transition from VR to AR, the OpenWire library had to be replaced with an MQTT library due to implications of the employed computing platforms, a standard gaming computer for the VR application and the Microsoft HoloLens for the AR application. This transition was easily possible simply by replacing the OpenWire library with an MQTT library, adjusting the MoM-related code in the AR application to use the new MQTT library, and enabling MQTT in the bowerick broker by adjusting a simple command line option. All other parts and components, such as exchanged messages, data format, de-/serialization, etc. could remain the same.

Furthermore, our development was eased by the functionality of bowerick to act as interactive Command Line Interface (CLI) client. The interactive CLI allows, e.g., to send and receive messages with all supported protocols, which helped in run-time testing and debugging of the system.

From our experience gained during the AR and VR studies, we consider the discussed approach very helpful. The development was not hindered by unnecessary limitations, conversions, or adapters. Instead, we could select the best tools, such as programming languages or libraries, for the particular use case, reduce boiler plate code, and focus on the actual functionality.

While we focused on development and integration effort, performance may still be an important factor. However, performance aspects should be considered on a case-by-case basis. For assessing performance aspects, bowerick offers benchmark functionality such as customizable message generators or a configurable benchmark consumer.

In future, we consider that MoM-based infrastructures will be increasingly used, e.g., EGS-CC uses Apache ServiceMix [13], which includes Apache ActiveMQ, as part of its infrastructure [14]. New developments at the German Space Operations Center (GSOC) consider the selection of the MQTT MoM protocol for integration [15]. CNES ISIS employs ZeroMQ, another MoM [16].

[1] Gad, R., "GitHub - ruedigergad/bowerick: Easing simple Message-oriented Middleware (MoM) Tasks with Clojure (and Java)," Online: https://github.com/ruedigergad/bowerick, last accessed 2018-06-13.
[2] The Apache Software Foundation, "Apache ActiveMQ – Index," Online: http://activemq.apache.org/, last accessed 2018-06-13.
[3] The Eclipse Foundation, "Jetty - Servlet Engine and Http Server," Online: https://www.eclipse.org/jetty/, last accessed 2018-06-13.
[4] Pivotal Software, "Getting Started · Using WebSocket to build an interactive web application," Online: https://spring.io/guides/gs/messaging-stomp-websocket/, last accessed 2018-06-13.
[5] Lombardi, A., "WebSocket, Chapter 4. STOMP over WebSocket," O'Reilly Media, Inc., Sept. 2015, Online: https://www.safaribooksonline.com/library/view/websocket/9781449369262/ch04.html, last accessed 2018-06-13.
[6] The Eclipse Foundation, "Eclipse Paho - MQTT and MQTT-SN software," Online: https://www.eclipse.org/paho/, last accessed 2018-06-13.
[7] MQTT Version 3.1.1. Edited by Andrew Banks and Rahul Gupta. 29 October 2014. OASIS Standard. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. Latest version: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.
[8] European Cooperation for Space Standardization, "Ground systems and operations - Telemetry and telecommand packet utilization," Online: http://www.ecss.nl/wp-content/uploads/standards/ecss-e/ECSS-E-70-41A30Jan2003.pdf, last accessed 2018-06-13.
[9] The Consultative Committee for Space Data Systems (CCSDS) "MISSION OPERATIONS SERVICES CONCEPT," Online: https://public.ccsds.org/Pubs/520x0g3.pdf, last accessed 2018-06-13.
[10] Gad, R., Valadas, M., Graf, H., Olbrich, M., Keil, J., Sarkarati, M., Bamfaste, S., Nicolini, F., Cowley, A., "Assessing the Potential of Virtual Reality for Improving ESA's Space Operations," SpaceOps 2018, May 2018.

[11] The Apache Software Foundation, "Apache ActiveMQ – OpenWire," Online: http://activemq.apache.org/openwire.html, last accessed: 2018-06-13.
[12] STOMP Spec Group, "STOMP - The Simple Text Oriented Messaging Protocol," Online: https://stomp.github.io/, last accessed: 2018-06-13.
[13] The Apache Software Foundation, "Welcome to Apache ServiceMix!," Online: http://servicemix.apache.org/, last accessed 2018-06-13.
[14] Schiller, T. M., "EKSE: A Command Line Interface for EGS-CC based Systems," SpaceOps 2018, May 2018.
[15] Gaertner, S. A., et al., "Rethinking Ground Systems: Supporting New Mission Types through Modularity and Standardization," SpaceOps 2018, May 2018.
[16] Gélie, P., Pasquier, H., Labrune, Y., "CNES Mission Operations Systems roadmap: towards rationalisation and efficiency with ISIS," SpaceOps 2018, May 2018.

Talks / **24**

Open Source DVB-S2 and DVB-S2X for GNU Radio

Author: Michelle Thompson[1]

[1] *Open Research Institute*

Corresponding Author: mountain.michelle@gmail.com

Phase 4 Ground is a project devoted to bringing an open source implementation of DVB-S2 and DVB-S2X to amateur radio terrestrial and space deployments. Any payload that complies with the air interface of 4-ary MSK FDMA up and DVB-S2/X+GSE TDM down will be able to take full advantage of a modern efficient open-source communications system. Phase 4 Ground is under active development and welcomes participation. Phase 4 Ground complies with ITAR 120.11 and publishes all work as open source.

Talks / **14**

Orekit, high end flight dynamics accessible to small cubesat teams

Author: Luc Maisonobe[1]

[1] *CS-SI*

Corresponding Author: luc.maisonobe@c-s.fr

Cubesat missions are often driven by small teams focusing on a single goal. There is generally only one instrument on-board, performing one specific measurement, in contrast with large Earth Observation platforms that hosts half a dozen instruments provided by several laboratories in an internationally coordinated project.

The small team main interest is gathering the data. Its members are often by themselves the end users of their own system, just as in open-source the developers of a program are users "scratching their own itch", as the saying goes. Flight dynamics is a necessary chore for the success of the mission, but not a goal by itself. There are generally no resources available for this topic, neither in terms of skills nor in terms of tools. As a result, many cubesat projects end up using the public TLE provided by SpaceTrack and some on-line path predictors to schedule their

connection to spacecraft for telemetry retrieval. TLE are notoriously low accuracy (it is a simplified analytical model, and measurements for a given object are not refreshed as often as the owner would like). This is sufficient (and widely used) for simple access prediction, but clearly not enough for more ambitious missions.

As cubesat missions address more and more needs, they reach a state where spacecraft location should be known with better accurary, for example to improve the geo-location of the measurements performed by the instrument, and therefore improving the scientific model resulting from the mission.

Orekit is a free software space flight dynamics library that has been available under the terms of the Apache License V2 since 10 years. It is widely used by many major space actors, including agencies, industry, research and academics. It is used for operational systems, mission analysis, studies and training. As an open-source library, it can be used to build any kind of system, from simple ones performing only prediction using TLE to highly accurate ones managing constellations with station keeping maneuvers and precise orbit determination. As cubesat now can have on-board GNSS receivers, it is possible for a cubesat project to perform its own orbit determination by downloading the data (either high level position-velocity or low level pseudo-range or carrier-phase) and processing it on ground with a small Orekit-based program. Orekit as a large community of space flight dynamics experts that can help smaller teams to set up their own systems.

This talk will present a selection of the features in Orekit that can be used for cubesat missions with limited resources.

Talks / 39

Cubesat Subsystems Preliminary Design: One Software Suite to Bind Them All?

Authors: Thibault Gateau[1]; Fabien Humeau[1]; Fabien Apper[1]

[1] *ISAE-SUPAERO*

Corresponding Author: thibault.gateau@isae.fr

Designing a nanosatellite requires close interrelation between different fields, with respectively strong level of expertise, all the more so as development progresses. During cubesats preliminary design many budgets are essential (mass, power, link, data, dissipation). Their local inputs, outputs and models are often intertwined. For instance, power budget rely on payload requirement, platform operational up-keeping, eclipses frequencies/duration, and batteries specifications. It will impact mass budget (e.g. number of required batteries, solar panels and wires, etc.), and reciprocally. But dissipation budget will also be concerned (batteries will only work between temperature extrema), heaters and radiators used will also impact mass budget and so on. All this process is, to our current knowledge, far to be unified with an ideal set of tools. Functionalities are often redundant between the different soft used, or even re-developed each time required rather than re-used. Fortunately some software bricks already exist, such as space mechanics libraries. Efforts on standardization are also undertaken (CSSDS) mainly concerning telecommunication protocols, and to a lesser extent, ephemeris formatting or equipment description. Even if they are paving the way for consistently interconnected suite of tools, proceeding end to end mission analysis lack of unified, consistent standards and open source tools.

Here come the idea for one tool to bind them all (and let's hope so not in the darkness bring them) or more precisely, a well defined suite of tools to help to get a strong consistency for a mission analysis preliminary design, which can follow the project to all it's live cycle thanks to a strong interconnection with experts tools. Optimally, each expert should be able to take (up to date) needed inputs on its own tools and provide to the team expected outputs, in a transparent way. Thanks to our previous and on-going nanosatellite projects, we have now a more practical vision on specific nanosatellite project needs, and redundant software usage and developments we are used to be facing.

That also in the scope of SUDOE Nanostar project proposition which aims at supporting training and development of student nanosatellites in Europe, where we are strongly involved (http://nanostarproject.eu/).

Talks / **26**

Testing and validation approaches for scientific software

Author: Juan Luis Cano Rodríguez[None]

Co-author: Nikita Astrakhantsev

Corresponding Author: hello@juanlu.space

Nowadays, even though software has a fundamental role in scientific research, the wide majority of scientists is primarily self-taught and received no formal training in software engineering, with often leads to quality and reproducibility problems1. The space industry is in a similar situation, with many incident reports describing "various aspects of complacency and a discounting or misunderstanding of the risks associated with software"[2][3].

One of the most useful engineering techniques, software testing, is also the one that presents the biggest gap between its perceived importance and the skill level of scientists in it[4]. Testing, as well as other good practices such as version control and code reviews, not only make code more reusable but also increase the productivity of the developer[5]. However, the special nature of scientific or algorithmic software makes it difficult to apply commonplace testing practices, since the challenges lie in "separating software bugs from model errors and approximation error"[4].

In this talk we will discuss some testing approaches (or lack thereof) present in scientific software that fall short in helping the developers find errors or increase their productivity, and propose some other strategies based on our experience with poliastro, an open source Python library for Astrodynamics[6]. These strategies make use of automated testing frameworks, help covering test cases in an exhaustive way, take advantage of analytical solutions of the problems at hand or public data when available, and guarantee self consistency when there is nothing to compare against. Finally, we will analyze the limitations of these approaches and discuss possible solutions.

References

1: Wilson, Greg, Dhavide A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven HD Haddock et al. "Best practices for scientific computing." PLoS biology 12, no. 1 (2014): e1001745.
[2]: Leveson, Nancy G. "Role of software in spacecraft accidents." Journal of spacecraft and Rockets 41, no. 4 (2004): 564-575.
[3]: Lutz, Robyn R. "Analyzing software requirements errors in safety-critical, embedded systems." In Requirements Engineering, 1993., Proceedings of IEEE International Symposium on, pp. 126-133. IEEE, 1993.
[4]: Hannay, Jo Erskine, Carolyn MacLeod, Janice Singer, Hans Petter Langtangen, Dietmar Pfahl, and Greg Wilson. "How do scientists develop and use scientific software?." In Software Engineering for Computational Science and Engineering, 2009. SECSE'09. ICSE Workshop on, pp. 1-8. Ieee, 2009.
[5]: Wilson, Greg, Dhavide A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven HD Haddock et al. "Best practices for scientific computing." PLoS biology 12, no. 1 (2014): e1001745.
[6]: Juan Luis Cano Rodríguez, Antonio Hidalgo, Shreyas Bapat, Nikita Astrakhantsev, Alberto Lorenzo Márquez, Pablo Rodríguez Robles, Ritiek Malhotra, et al. 2018. "Poliastro/poliastro: Poliastro 0.9.1 (pickle Edition)". Zenodo. doi:10.5281/zenodo.1245650.

Talks / **27**

Ancillary Data Production for SmallSats with SPICE

Author: Marc Costa Sitjà[1]

[1] *RHEA System for European Space Agency*

Corresponding Author: mcosta@sciops.esa.int

SPICE is an information system the purpose of which is to provide scientists and engineers the observation geometry needed to plan scientific observations and to analyse the data returned from those observations. SPICE is comprised of a suite of data files, usually called kernels, and software -mostly subroutines-. A customer incorporates a few of the subroutines into his/her own program that is built to read SPICE data and compute needed geometry parameters for whatever task is at hand. Examples of the geometry parameters typically computed are range or altitude, latitude and longitude, phase, incidence and emission angles, instrument pointing calculations, and reference frame and coordinate system conversions. SPICE is also very adept at time conversions.

This contribution will outline how can SPICE be used for CubeSats to build up a cheap, efficient and robust system to obtain ancillary data.

Talks / **10**

CubeSat, 3D Printing & COTS

Authors: Martín Ezequiel Molina [1]; Hernán Paez [1]; Marco A. Reyna[2]

Co-author: Talia Bessone [1]

[1] *Universidad Tecnológica Nacional Facultad Regional Córdoba-Argentina*

[2] *Comisión Nacional de Actividades Espaciales (CONAE)*

Corresponding Authors: martinezequielmolina@gmail.com, hernanpaez1@gmail.com, malvarezreyna@vengsa.com.ar, taliabessone@gmail.com

General Objectives
Encourage the regional development of space applications for social-economic purposes based on CubeSat technology.
Specific Objectives
To foster the approach between the academic field and the aerospace environment, particularly to undergraduate students. To conceptualize, design, produce, test and start a set of CubeSats with academic purposes making use of the FDM 3D printing technology and Commercial Off-The-Shelf (COTS) components, maximizing the use of the state of the art of the involved technologies, focusing the efforts in the development of interfaces, software, systems integration and above all in the development of an application-proposal to fill a not only locally but regionally vacancy area.

Articulation and short-term implementation and use
Encourage the professional and academic articulation of the project. Call local and regional institutions interested in CubeSat standard-based Smallsat technology, for a hands-on training with an academic approach, by means of the developed 3D Printing & COTS CubeSat engineering model replicas. A general block diagram showing the complete architecture of a mission based on CubeSat can be seen on Fig.2. Creating these type of satellites aims to train and attract undergraduate students towards science and technology in aerospace matters, focussing efforts on satellital standards growing worldwide. Therefore the CubeSat replicas will be built with 3D printing fused deposition modeling and COTS electronics, based mainly on Open-source hardware (e.g. Arduino).

Strengths
- The group currently working on this project, has been developing activities linked to the robotics field with educational purposes in High Schools and Colleges.
- Availability of 3D printers.
- Self-funding network for minor developments.
- Technical coaching in aerospace matter and access to GomSpace and ISIS CubeSat specs (from CONAE Academic Laboratories).

Technology involved
The Arduino platform was chosen for the development due to its low cost, local availability and work philosophy. The 3D Printing & COTS CubeSat project include:
- On board Computer (Arduino Mega 2560).
- One camera module (C439).
- Digital Accelerometer (ADXL345).
- Light sensor (TSL2561).
- NetWork Communication module (NodeMCU,NRF2104).
- Up/Down Link RF module.
- Solar Panels (18V/40mA).
- Expansion boards FFPC104.
- Battery module (ICR18650 pack).

FDM 3D printing technology is employed for the structure development, achieving a CubeSat replica for academic practice. The structure can be seen in Fig.3 wherein the modules involved are exposed.

Space Segment
The space segment is composed of 6 general subsystems:
- Energy subsystem (EPS): its implemented with a rechargeable cell array ICR 18650 (3.7v/4800mAh) they supply 7.2V/9600 mAh to the system. A solar panel array collect the energy to charge the batteries by a TP4056 module.
- Communication subsystem: The Cubesat Network use a NodeMCU (master hub) and RF24L01 Wifi receptors, to the communication between each space segments. One configured as Master Hub and the others as Slaves.
- Telemetry, telecommand and (TT&C) control: The data download and remote control from the ground station use RF transmission modules working at 433MHz. Also it has a radio beacon which transmit the EPS status at 144Mhz to isolate it from the principal frequency.
- Position Subsystem: To know the orientation of the cubesat and maintain the direction of the antennas always to the Earth and the solar panels to the sun, use a 3 axis one accelerometer and one light sensor.
- Payload: A 1.3 Mp camera module, it's the principal payload.
- OBC: Use a Mega 2560 Arduino Board to check, manage and store the data from all subsystems.

Ground segment
This segment simulate a real Ground station. Consist in:
- RF(433 Mhz) reception module with antenna.
- Nano Arduino Board integrated to the receive data from the space segment (downlink) and remote control (uplink) to make manoeuvre.

Mechanical Model
The Space Segments have 2U standard cubesat unit. Each are build using plastic by FDM 3D printing technology and L shape aluminium profile all attached with M3 threaded rod, bolts and nuts (Fig.4). That material mixture provides greater mechanical rigidity than use only plastic like previous versions. In this way an educational engineering model similar to one for commercial purposes is achieved.

Firmware
Currently running with V1 version that integrates all the involucrated modules in the subsystems mentioned previously.
This version provides a minimum necessary code to obtain through ground segment, orientation, energy status and the cubesat beacon. The V2 version is being developed, it will be incorporate the telecomands, general system optimization and camera manipulation for images acquisition (payload).

Talks / 6

HyperSat – new standard for open hardware/software microsatellite platform – phase B1 – first tests of the laboratory models.

Author: Marcin Stolarski[1]

Co-authors: Tomasz Zawistowski [1]; Martyna Ulinowicz [1]; Szymon Mateusz Sipa [1]; Szymon Wasilewski [1]; Marcin Bieda [1]; Radosław Ławrynowicz [1]; Daniel Kaczmara [1]; Michał Kuklewski [1]; Arkadiusz Wójcik [1]; Paweł Czapski [1]; Paweł Kowalczyk [1]; Maciej Kuźniarski [1]; Tomasz Górski [1]; Marcin Mazur [1]; Tadeusz Kaluba [1]; Paweł Zienkiewicz [1]; Grzegorz Kasprowicz [1]; Patrycja Turek [1]

[1] *Creotech Instruments S.A.*

Corresponding Author: mstolars@gmail.com

HyperSat MINI (HSN) project was set up with an intention to develop a new satellite platform which would facilitate space mission preparation and implementation in shortest time possible. Satellites will be offered in a modular format – the smallest module size being 35x35x10 cm and having a mass of up to 10 kg. The structure will be scaled up to the limit size of 35x35x60 cm and the mass of 60 kg. All necessary satellite subsystems such as power system, on-board computer or RF communications system (using S-band and X-band up to 50 Mbps) will be designed and built with new standardization in mind. All subsystems will be connected by HyperSat Data Network (HDN) which is compatible with Space VPX and SpaceWire standards. The main advantage of the platform lies in its modularity and repeatability so that the customer can freely configure the satellite according to their needs by using ready-made, off-the shelf modules. The philosophy behind the development of HyperSat is in analogy to a cubesat project, although HyperSat is designed for satellites of larger sizes and mass. An upcoming platform is to be backward compatible with cubesat modules. It is proposed that all satellite development comprising its structure, electronic systems and software is to be made available for free under an open source license (Open Hardware License, Open Software). The HyperSat project is funded by the Polish National Centre for Research and Development and led by a private company Creotech Instruments S.A. New partners are welcome to join the development of a new bus standard or modules compatible with it. Current status of the project is as follows: Phase A (Platform Definition) has been completed and work concentrates on Phase B (Platform Design).
At the beginning of Phase B1 first prototypes (laboratory models) and simulations of satellite subsystems ware created. This paper presents the results of initial test.

Posters and Demos / 7

Full Stack Data Science: Using Python to download, clean, analyze and visualize Gaia data

Authors: Juan Luis Cano Rodríguez[None]; Roberto Izquierdo Amo[None]

Co-author: Mariona Badenas Agustí

Corresponding Authors: hello@juanlu.space, robertoia1987@gmail.com

Python is not only one of the most important programming languages in the software industry1[2], but also shows the most solid growth among the widely used ones thanks to its popularity in the Data Science field[3][4]. Python gained lots of traction in the scientific computing field in the 90s and the 2000s thanks to its simplicity and its power as a "glue language", allowing scientists to unify their tasks and use only one tool[3]. However, with the explosion of the Big Data movement in the 2010s and the evolution of web browsers and personal computers, it was no longer possible to use one single language for Data Science: efficient data access has to be done through databases using SQL, all the Hadoop stack is written in Java or Scala, while modern visualization is usually developed in JavaScript. Even if some current solutions, such as PySpark, enable the use of Python for Big Data, the deployment is often complex, the workflow fragile, and a gap between Small and Big Data still exists.

We argue that the aforementioned complexity is an obstacle for scientists without formal training in Software Engineering and also for the general public, particularly in Astronomy. In recent times though, the Python ecosystem has evolved to catch up with the Big Data world, and new tools have appeared to process, analyze and visualize big amounts of data (where we will freely define "big" as "not fitting in RAM", i.e. in the order of tens of gigabytes or more). On the one hand, libraries like numba allow Python to scale up by compiling a subset of the language to assembly code in a Just-in-Time fashion (hence improving the performance of single node programs), and on the other hand projects like Dask are bringing distributed and out-of-core capabilities to the usual "small data" tools of the Python in Data Science stack (numpy, pandas, scikit-learn), allowing easy reuse of existing codebases and easy scaling to tens to thousands of nodes with minimal changes in the deployment.

In this talk we will showcase the power of these modern Python packages to perform Full Stack Data Science on the Gaia Data Release 2 (DR2), a recently released astrometry dataset with observations of 1.3 billion stars by the Gaia mission. We will demonstrate how to use Pyia and Astroquery to easily download Gaia data[4][5], how to process big amounts of data on a modest laptop using Dask, and how to scale that to a distributed cloud computing cluster to increase performance with minimal cost and minimal code changes. We will also feature special Python libraries for Big Data visualization that avoid common visualization pitfalls such as Datashader, Holoviews and Bokeh. To finish the presentation, we will justify how these tools simplify the access to Astronomical data in general, discuss the limitations of the approach and talk about future developments.

References

1: TIOBE index https://www.tiobe.com/tiobe-index/
[2]: GitHut https://madnight.github.io/githut/#/pull_requests/2017/4
[3]: The Incredible Growth of Python https://stackoverflow.blog/2017/09/06/incredible-growth-python/
[4]: KDNuggets Poll https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html
[3]: Beazley, David M. "Scientific computing with Python." In Astronomical Data Analysis Software and Systems IX, vol. 216, p. 49. 2000.
[4]: Adrian Price-Whelan. "Adrn/pyia: V0.3". Zenodo, June 5, 2018. doi:10.5281/zenodo.1275923.
[5]: Price-Whelan, A. M., B. M. Sipőcz, H. M. Günther, P. L. Lim, S. M. Crawford, S. Conseil, D. L. Shupe et al. "The Astropy Project: Building an inclusive, open-science project and status of the v2. 0 software." arXiv preprint arXiv:1801.02634 (2018).

Posters and Demos / **19**

ISTsat-1 Control Protocol (INCP)

Author: Alexandre Silva[1]

Co-author: Rui M. Rui M. Rocha [1]

[1] *Instituto de Telecomunicações*

Corresponding Author: istsat-one@tecnico.ulisboa.pt

ISTsat-1 is a CubeSat currently under development by students and faculty from Instituto Superior Te´cnico (IST) along with amateur radio experts from AMRAD. As with any spacecraft, it will need to be monitored, controlled and results from scientific experiments have to be retrieved. How these tasks are communicated between the ground and the spacecraft is the domain of the ISTsat-1 Control Protocol (INCP) which was designed to serve as the high level protocol for telemetry retrieval and overall control of the ISTsat-1.
The INCP follows a client-server approach with asynchronous message-passing. In this approach, the Ground Segment is the client and the spacecraft is the server, namely the subsystems that compose the spacecraft, which are computationally independent. Additionally, INCP is also used for inter- subsystems communication inside the spacecraft.
Overall, there are four groups of operations covered by INCP: data retrieval, event reporting (such as errors), command execution and diagnostic execution.
Data retrieval is used for transferring produced data from the satellite's subsystems to the ground.

Each data type is unique and can be either periodic or singular. Most of the data is periodic, such as the scientific telemetry while for some housekeeping information only the instantaneous values are relevant.
Event reporting is, for the most part, used to report the occurrence of some predetermined events, typically errors. These are produced infrequently and most can have no value at all. The most important information is that they occurred.
Each subsystem has a set of commands which can be executed if so instructed by the ground. Each command is identified by a unique opcode and has input/output values. These arguments and values, are specified by the subsystem and have well defined types. Furthermore, commands as well as diagnostics are only executed by the spacecraft if properly authenticated. Critical messages carry a HMAC (Hash Message Authentication Code) to ward of against unauthorised use of the satellite. Data and event reports don't have any such mechanism since they don't perform any significant or critical operation on the satellite.
The goal of diagnostics is to actively test a component and verify its status. Once issued, the resulting test can take a while to be performed and may even finish after the connection window with the ground station has closed. In such occasions, the results are retrieved on the next window. These results are similar to command results but also include the start and end timestamps of the diagnostic execution.
The protocol was designed with a few core concepts in mind in order to best fit the context of the ISTsat-1 mission.
Firstly, all messages can be interpreted in isolation. In other words, one must not have to know what messages were previously exchange in order to "understand" a received message.
Secondly, it's assumed that the underlying protocol stack guarantees several properties. Specifically, message loss recovery, duplicate message repudiation, ordering and integrity.
Finally, due to the bandwidth requirements Type-Length-Value type formats such as ASN-1 were avoided, especially text based ones such as XML. Thus, a binary static format was chosen were the type of the message is deduced from it's opcode, unique per type of message.
The protocol's core is already implemented with much of it's components tested individually. However, and in the context of the ISTsat-1 project, the modules which describe the various commands, data, etc of the spacecraft subsystems still need to be completed. Furthermore, the protocol and its implementation are yet to be tested in a live environment.

Welcome and Keynote / 49

Welcome and Keynotes

Posters and Demos / 1

Docker Containers for Open Source Space Software

Author: Artur Scholz[1]

[1] *LibreCube*

Corresponding Author: artur.scholz@librecube.net

Programming great software is one thing - to deploy it is a complete different thing. By deploying, we mean the installation or setup of the software on a different machine from where it was developed. Software deployment is necessary for production shipment but also for testing and collaboration in the same development environment.

All too often, this is a painful and unrewarding process. Commonly numerous implicit and explicit dependencies must be fulfilled by the target environment. For example, a SQL database may have to be installed or a specific version of a Python interpreter must be made available. Usually the developer would just expect the user to operate the software in the appropriate environment; but to get to this set up is often cumbersome and non-trivial task.

Docker comes to the rescue! Docker is a computer program that performs operating-system-level virtualization also known as containerization. Docker is a tool that can package an application and its dependencies in a virtual container that can run on any Linux server. Think of it as a light-weight virtual machine. It is based on Linux containers but provides a user-friendly abstraction to it, making its usage extremely simple.

Docker containers can be a useful method for building up space mission ground systems in terms of a micro-service architecture. For example, a mission control system is usually compose of various software components for telecommand preparation and telemetry checking, as well as automation and data storage. Building these components as Docker containers helps to allow for scaling and redundancy management, and an overall modularisation of the system.

This presentation will introduce the basics on how to create and run Docker containers. Using the example of an open-source NoSQL database with an REST API as developed by the author, a practical inside into the syntax and quirks of Docker will be given. Finally, a number of dockerized open source projects for use in CubeSat missions will be shown.

Posters and Demos / 3

Development of a flight software framework for student CubeSat missions

Authors: Olman Quiros-Jimenez[1]; Duncan d'Hemecourt[2]

[1] *Costa Rica Institute of Technology*

[2] *George Washington University*

Corresponding Authors: olmanqj@gmail.com, ddhemecourt@gwu.edu

Abstract
During 2018 a student CubeSat project was developed at George Washington University (GWU) for the first time. The small satellite mission implemented a software stack with the hope of creating a simple and lightweight framework for future academic CubeSats. The developed flight software consisted of a collection of fundamental services and an application layer, which was executed above the Network Layer and the Operating System. Accompanying ground station software was also developed for the mission. This paper presents the resulting software framework, its simple architecture, features, software quality attributes, and the decisions made during the design and implementation. The paper will address and compare other available open source software frameworks for CubeSat missions and will propose a general architecture for any CubeSat mission at an introductory level. This generic framework will define the minimum features and standards to obtain a flexible, portable and reusable software library. The paper will provide students without previous CubeSat experience some initial information and examples to start the development of a CubeSat flight software.

Posters and Demos / 11

High-performance on-board computer, data handling and SDR platform for cubesats

Authors: Juan Ramos-Castro[1]; Josep Colomé[2]; José María Gómez[3]; Jordi Portell[4]; Adriano Camps[1]; José Bosch[5]; Manuel Carmona[5]; Albert Casas[6]; Lluís Gesa[2]; Juan Mauricio[7]; Juan Francisco Muñoz[8]; David Roma[6]; Carles Sierra[2]

[1] *UPC / IEEC*

[2] *IEEC / CSIC*

[3] *UB / IEEC*

[4] *IEEC / ICCUB*

[5] *UB*

[6] *UB / ICCUB*

[7] *ICCUB*

[8] *UPC*

Corresponding Authors: jportell@fqa.ub.edu, gesa@ieec.cat

Built upon the successful expertise of LISA Pathfinder, the 3Cat series cubesats (from UPC NanoSat Lab), Solar Orbiter and Gaia, the Institute for Space Studies of Catalonia (IEEC) is designing and implementing a high-performance multipurpose platform for cubesats that can be adapted to different commercial and scientific uses. It provides a robust on-board computer with redundancy to control the spacecraft state and telecommands, a versatile software-defined radio based on a high-end FPGA SoC providing high-speed downlink capabilities, a powerful on-board data handling system, and an efficient on-ground telecommand and basic data handling framework. This solution will push the cubesat concept to its limits, allowing to achieve performances for which larger-sized missions would be required otherwise. We present the overall features of this platform, its capabilities, and some possible use cases.

Posters and Demos / 15

Using AI onboard of small satellites

Author: Xabier Crespo Álvarez[1]

Co-author: Redouane Boumghar [2]

[1] *Embedded Systems Engineering Consultant*

[2] *Research fellow at the Advanced Missions Concept and Technology Team, ESA/ESOC*

Corresponding Author: xabi@crespum.eu

The number of cubesats and small satellites sent to orbit is rapidly growing. Not only that, but the market trends toward the launch constellations of hundreds of satellites in LEO. This situation will become very challenging in terms of operations and data management for all the New Space companies. In order to overcome these problems, we need to increase the level of autonomy of satellites so that they don't depend on regular human intervention to function neither on a ground to space telecommunication link. These factors are generally very hard to scale.

During the last years, we've witnessed big advances in the so called edge computing industry, which aims to give unprecedented computational power to devices in constrained environments. These devices are limited in many aspects, including power availability and/or connectivity, just as satellites are. The ultimate goal of edge computing is to reduce their dependency on a central node to extract and analyse the data gathered.

If we consider ground stations as the central nodes of constellations, and satellites the end devices, we can leverage these techniques for using them into space. There are many benefits in doing so. We are particularly interested in one of them, the usage of artificial intelligence onboard of small satellites. This option has already been explored in the past by researches, but only now is becoming a reality for every company. Cubesat missions that embed high performance computing devices are already planned to test these devices in Earth orbit.

The availability of AI onboard can help small sats reach a new level of autonomy. Satellite operations can be simplified by using it to identify complex time-series or to detect anomalies that could harm the satellites and act accordingly without needing to download all the telemetry. There's also a lot of potential for applications that use a big amounts of data such as Earth Observation. Thanks to AI we could move part of the application layer that is currently being executed on the ground to the space segment. The images can be analysed onboard and we could download just the results. This would save a lot of bandwidth as well as help up with frequency coordination.

These are just two straight forward examples, but there are more. AI can also be extremely helpful on deep space missions or even for the autonomous coordination of constellations. It's an option worth being explored.

We found really difficult to compare edge computing tools on paper. Each manufacturer takes a different approach to describe their specifications, so we decided to do it empirically. We took three different COTS solutions (an NVIDIA Jetson TX1, an Intel Movidius and a library for microprocessors using Mbed OS called uTensor) and compared them in terms of accuracy, power consumption, processing time and other parameters.

For the comparison to be fair we created an AI model and exported it to the three platforms. Then we created a set of inputs and fed the model with them to infer the outputs. During this process we monitored the parameters under test. The results is what we will be presenting in this conference.

Posters and Demos / 16

FACT, An open project for SmallSat building capacity

Authors: Samer Lahouar[1]; Kamel Besbes[1]

[1] *CRMN*

Corresponding Author: samer.lahouar@crmn.rnrt.tn

The emergence of experimental nanosatellites and especially CubeSats for space exploration and experimentation, gives low-cost access to space technologies to emerging countries, such as Tunisia. In fact, CubeSats can be easily made since their components are widely available. Moreover, CubeSats have versatile applications in different domains, such as: communications, remote sensing for agriculture and territorial management, border and environmental risks surveillance, geolocation, etc.
On the other hand, the majority of space applications are within the scope of the National Research Priorities in Tunisia.
In fact, every space application involves one or more national priorities. As an example, space technologies in general and CubeSats in particular would help in: the digital and industrial transition (CubeSat prototypes and systems); securing energy, water, and food (space surveillance); Education (new profiles); etc. The space segment would also help in the implementation of new smart applications, such as smart cities, smart agriculture, smart transportation, iOT, etc., without the dependence on other countries.
For all these reasons, it is very important for Tunisia to start its first space project by making and testing CubeSats, which have a high Technology Readiness Level (TRL). This would also provide important capacity building in space technologies for students and engineers, which in turn would help in creating new jobs for new profiles. The project is based on open source software and hardware.
The FACT project : Fabrication and Application of CubeSats in Tunisia, Conducted by 4 partners: Research Centre, Academic Research Lab., Robot Compagny and National Centre for Earth Observation, would allow different specific objectives, such as:
• Acquiring the important concepts of space technologies having high economic added value applications, at relatively low costs.
• Capacity building of engineering, masters, and Ph.D. students in emerging technologies.
• Management of the research collaboration between the members of the consortium composed of a research center, a higher education and research institute, a private company, and a research center belonging to the Ministry of Defense.
• Implementation of CubeSat mounting and testing facilities that can be used for collaboration on other industrial aerospace projects (national or international).
• Help in the implantation of new aerospace companies (national or international) in Tunisia by providing qualified engineers in space technologies.
• Contribution to reinforce national Research and Innovation activities related to space technologies in accordance to different studies (OCDE, BERD, UN, ESCWA, and UNOOSA).
The FACT project would generate the following results:
• Acquisition and installation of high tech equipment that would allow the fabrication and testing of CubeSats in collaboration with the project partners.

• Upgrade the technical levels of researchers and engineers of the different consortium members by introduction of space technologies.
• Training on the "System Approach" used in developing a space mission (NASA documents)
• Research development of new electronic embedded systems and modules rated for space missions and space use
• Fabrication of the first Tunisian CubeSat prototype using national skills, which will give access to important international financing and collaboration on bigger projects.
• Help in the implantation of the space industry in Tunisia (in the Sousse Technopark in particular) and the creation of new startups for the development of space modules and systems.

Posters and Demos / 20

Distributed Delay Tolerant Protocol

Author: Pedro Gameiro[1]

Co-author: Rui M. Rocha [2]

[1] *Instituto Superior Tecnico*

[2] *Instituto de Telecomunicações*

Corresponding Author: istsat-one@tecnico.ulisboa.pt

A cubesat is a very small, low cost, artificial satellite designed for space research purposes, very popular in the academic community. Cubesats are normally deployed in Low Earth Orbit (LEO), which can be defined as orbits up to 1000 km above the earth surface, with an orbital period of approximately 90 minutes at a speed of 7.8km/s. However, a big consequence of this, is that a particular ground-station only has a very small time-window of line-of-sight to communicate with the satellite. Offering even more obstacles, space-link communications are very unstable, error prone, and of low debit. The challenges in space-link communications are not traffic congestion (like on earth), but long propagation delays and high bit-error rates. This motivated the creation of Delay/Disruptive Tolerant Network (DTN) protocols, a concept designed to deal with the characteristic problems of disruptive environments. However, even with DTN, the transmission of big data files can be difficult. This is specially true in the case of cubesats, working typically in Low Earth Orbits (LEO), which suffer from very long disruptions periods with the Ground Stations (GS) on Earth.
The ISTsat-1 is the first cubesat being developed at the Instituto Superior Tecnico (IST), with a 1U size (103cm cube). For this project we intend to create an enhanced solution. A new approach that will tackle this difficulties in a new, more powerful way. To create a distributed protocol capable of expanding a single transmission session over several links and hosts (ground-stations in this particular case). This means that the satellite, after starting a transmission with a given Ground Station (GS), leaving its line-of-sight and consequently disconnecting the link, does not need to perform a full orbit to resume the transmission. It can just continue the transmission with the next GS available in its orbit. This can result in a very significant performance enhancement, both due to the very limited available time-windows and the typical characteristics of space communications, like high delays and low throughputs.
A particular interesting use case, that may be possible with this distributed approach, is the enabling of communications between clusters of cubesats. Since these clusters are randomly deployed in an area of interest, communications links are not constant. The flexibility of a distributed approach allows to compensate for this, since there is no fixed route or peer through which a transmission must go.

Posters and Demos / 21

Open Mission Control Technologies

Author: Jay Trimble[1]

[1] *NASA*

Corresponding Author: jay.p.trimble@nasa.gov

Open Mission Control Technologies (Open MCT) is NASA open source software, available on GitHub, with an Apache II license. It brings mission control data visualization capability to the desktop, tablet and phones, using web browsers. Open MCT is currently in use on at least three operational CubeSat missions, including the Mars Cube One (MarCO) missions and the Arcsecond Space Telescope Enabling Research in Astrophysics (ASTERIA).

The key features of the platform are data visualization, all of your data browseable and searchable in one integrated environment, user composition of displays, integration with multiple data sources, and modularity for customization to different mission requirements. The code is available on GitHub at https://github.com/nasa/openmct. An informational website with documentation, tutorials and an online demonstration is available at https://nasa.github.io/openmct/.

In this demo we will show the capabilities of Open MCT, using data from a simulated lunar rover mission to showcase the capability for missions. The displays will show data from multiple domains in one integrated environment, including telemetry, imagery, mission timelines, lunar traverse maps, science data, science notebooks with integrated data, and logic driven summary widgets. We will demonstrate the assembly of this data into displays by users, without programming.

Posters and Demos / **25**

VIGIL Radar Sentry

Author: Michelle Thompson[1]

[1] *Open Research Institute*

Corresponding Author: mountain.michelle@gmail.com

VIGIL is a small radar in lunar orbit to track debris from deep space launches. There have been dozens of rocket stages in lunar-crossing high Earth orbit. There is a possibility that some have been captured in lunar orbit by 3-body effects. With increased travel to the Moon, unexpected objects in lunar orbit pose a risk. Since these objects are not large, and specular flashes may be intermittent, a radar must both maximize power on the target and integrate for a long time. Linear frequency-modulated (LFM) continuous wave (LFMCW) is a method that would address these needs and also avoid range ambiguities.

Posters and Demos / **28**

Out of the box tools for SmallSat operations with SPICE

Author: Marc Costa Sitjà[1]

[1] *RHEA System for European Space Agency*

Corresponding Author: mcosta@sciops.esa.int

SPICE is an information system the purpose of which is to provide scientists and engineers the observation geometry needed to plan scientific observations and to analyse the data returned from those observations.

The ESA SPICE Service and NAIF provide a suite of tools that using SPICE data provide ready-to-go functionalities that have proven to be operational for Planetary Missons. Some examples are WebGeocalc, SPICE-enhanced Cosmographia and a set of Python packages such as spiops. This contribution will provide a live demo on how these tools could be applied to SmallSat operations.

Posters and Demos / 29

Open-Source Mission Planning Tool for University Satellites

Authors: Markus Grass[1]; Marc Costa Sitjà[2]; Jonas Keim[1]

[1] *University of Stuttgart*

[2] *RHEA System for European Space Agency*

Corresponding Author: mgrass@irs.uni-stuttgart.de

During the last decade the amount of Cube-Sats and small satellites from Universities all around the globe increased rapidly. One of these small satellite, the Flying Laptop (FLP), is operated by the Institute of Space Systems (IRS, University of Stuttgart). The operations team uses third party software besides self-written pipelines to propagate the satellite orbit. Without hosting all the source code itself it might be difficult to adjust/implement additional functionalities.

Following an internship at the European Space and Astronomy Center (ESAC/ESA) an in the context of a master thesis, a mission planning tool (MPT) was developed using the well documented open-source library SPICE, SPICE-enhanced Cosmographia by the Navigation and Ancillary Data Facility (NAIF/JPL) and the open-source General Mission Analysis Tool (GMAT/NASA Goddard) which are also used by NASA and ESA themselves. Functionalities implemented include but are not limited to: Creating a data base out of GPS or NORAD TLE and attitude telemetry for analisis purposes (SPICE), propagating orbits and attitudes for short term mission planning (up to 1 month) including pointing to self-defined targets or events (SPICE/GMAT). Visualization for most of this data is also included (Cosmographia). At this moment the MPT is aligned with some needs of the FLP operations team to serve as an addition to already existing software to provide redundant output for some usecases. One of the major advantages of this modular based code is the high adaptability to new requirements. The code for this project is open source and is available as Pyhon Package Index Package and on Git from BitBucket server.

This contribution will point out some capabilities of this mission planning tool and hopes to rise the interest of others to develop this project further. The ultimate goal is a reliable open-source mission planning tool which could be adjusted to any mission within hours and maybe even setting a standard for mission planning tools among universities.

Posters and Demos / 32

ASTRE/Tolosat abstract

Author: Javier Navarro Montilla[1]

[1] *ASTRE*

Corresponding Author: j_navarr@etud.insa-toulouse.fr

ASTRE (students' space research association of Toulouse) is a student space association run by university students in collaboration with professors and the CSUT (University space center of Toulouse). The main objective of ASTRE is to develop student interest for space science and technology by developing engineering projects. ASTRE also aims at becoming an organization where students from Toulouse could share their knowledge of space with other students.

The ASTRE's first project is a 2U Cubesat called Tolosat. 30 students from 4 different schools and various fields work in this project. The main objectives of Tolosat are:

1. Prove that GNSS-r (Global Navigation Satellite System – reflectometry) altimetry can be realized by a 2U CubeSat:
   Earth is bathed in GNSS signals. Usually, these signals are exploited for geo-

localization. However, GNSS-r technology offers using GNSS reflected signals to have information about the Earth surface (soil humidity, roughness …). Tolosat aims at using GNSS-r technology for topographic mapping of the Earth surface using Galileo signals. The concept of this mission is based on Valery U. ZAVOROTNY's paper called "Tutorial on Remote Sensing Using GNSS Bistatic Radar of Opportunity". A first analysis shows that large surfaces of plane water, like the lakes, will increase the signal to noise ratio (SNR) of the reflected signal making it a good option for GNSS-r altimetry demonstration on a Cubesat.

2. Establish an SSH (Secure Shell) link with IRIDIUM satellite constellation in order to communicate with Tolosat from Earth at any moment when the spacecraft is in range of the IRIDIUM constellation:
   One main problem of Earth communication with satellites is the localization of the ground stations. There are some Earth places, like the oceans, where large antennas cannot be deployed in order to communicate with the satellites. As a consequence, Tolosat aims at proving that communication with a satellite can be established using IRIDIUM constellation as a relay.

Tolosat project is currently on phase A: a feasibility study is being developed. As sharing the knowledge is one of the first objectives of the ASTRE association, open source solution are being studied for the design of Tolosat space mission. Libre Space Foundation open source ground stations are being analyzed by the mission analysis team as well as other possibilities of open source hardware and software.

Posters and Demos / **33**

Novel Design of the Functional Structure of an 8U CubeSat to be 3D printed in thermoplastic

Authors: Jonathan Becedas[1]; Andrés Caparrós[1]; Gerardo Rodríguez[1]; Antonio Ramírez[2]; Pablo Morillo[2]

[1] *Elecnor Deimos*

[2] *Elecnor Deimos Satellite Systems*

Corresponding Authors: jonathan.becedas@elecnor-deimos.es, andres.caparros@elecnor-deimos.es, antonio.ramirez@deimos-space.com, pablo.morillo@deimos-space.com

Additive manufacturing is changing the way of designing structural parts. It offers the following benefits: design for need instead of design for manufacture paradigm, fast prototyping, reduced environmental impact, process optimization, mass reduction and applicability to new materials, among others.
In the space sector mass reduction is a major challenge. Structural components require the use of materials which have to resist very demanding conditions during launch and during operation in orbit. In addition, those materials shall have reduced mass in order to reduce launch costs. The introduction of new materials such as termoplastics with high resistance and low density can dramatically reduce the total mass of a spacecraft by fulfilling the requirements provided by launching authorities and by guaranteeing operability in the space environment.
In this work, a novel design of an 8U Cubesat structure is proposed to incorporate the benefits of additive manufacturing in thermoplastic polyetherimide (PEI) with the objective of qualifying it to be used in Low Earth Orbit. The design incorporates the lessons learned from the mechanical qualification tests and analysis of a previous structure in PEI proposed by the research team.
This study is being part of the H2020 European Project ReDSHIFT (Project ID 687500).

Posters and Demos / **34**

The Earth's micro-meteoroid environment observed by a dust sensor on-board a CubeSat operated by students

Authors: Nicolas Altobelli[1]; Marc Costa[1]; Ralf Srama[2]

[1] *ESA*

[2] *University of Stuttgart*

Corresponding Author: nicolas.altobelli@sciops.esa.int

We propose to explore the near Earth micro-meteoroid environment from a CubeSat platform, using a ‚Dust Telescope' (DT) operated by students as science payload.

The DT, developped by the Aerospace Enginneering Department of the University of Stuttgart is an in-situ dust sensor aiming at retrieving simultaneously the orbital elements and chemical composition of particles in the sub-micrometer to sub-millimeter size range. The DT is made of two main sub-systems: an electrostatic charge sensor at the instrument entrance, able to determine the impact velocity vector of the particles entering the instrument; and an impact time-of-flight mass spectrometer, able to constrain the particles elemental composition. This type of in-situ instrument, combining the dynamical and compositional information of the same particle, has never been flown thus far, but ready-to-fly versions, scalable for different mission concepts, have been built and tested at the University of Stuttgart. The CubeSat platform to host the science payload, is design and developed by the Universidad Europea de Madrid.

This mission would build up on collaborations with various european universities and on the education and outreach activities performed at ESAC in recent years, which have led to the design and set-up, by students, of an antenna and an operation room taylored for CubeSat operations.

Posters and Demos / **35**

Assessment of the influence of mechanical joints on thermal conductance between structural elements in the small-satellite thermal model.

Authors: Patrycja Turek[1]; Piotr Furmański[2]; Piotr Łapka[2]

[1] *Creotech Instruments S.A.*

[2] *Warsaw University of Technology*

Corresponding Authors: patrycja.turek@creotech.pl, pfurm@itc.pw.edu.pl, piotr.lapka@itc.pw.edu.pl

HyperSat is a new, modular microsatellite platform for space missions to be realized in a short time and at a low cost. It can support missions of a single satellite or constellation of spacecrafts. Currently, the HyperSat platform is under development. One of its crucial components is Thermal Control Subsystem.
In the space vacuum environment convection does not take place. Hence, the generated heat is transferred by conduction to the outer surfaces of the satellite structure and radiated to the surrounding space. Because of surface imperfections, micro gaps may occur between two contacting elements leading to decrease in the heat transfer. Therefore, understanding of the thermal conductance in the space environment as well as the proper assessment of its impact on the satellite thermal model accuracy in comparison to the physical model is of a prime importance. The objective of this paper is to examine the influence of mechanical joints on the thermal conductance and to describe the methodology of its study. Furthermore, a design of the laboratory test, on which the conductance validation tests will be carried out, is presented. The topics tackled in the paper will be further developed and the resulting conclusions will be considered in the final Structural Thermal Model (STM) design of the HyperSat platform.

Posters and Demos / **42**

Open source flight computer for model rockets

Authors: Dominik Marszk[None]; Claudiu Cherciu[None]

Corresponding Author: dmarszk@gmail.com

Model rocketry has inspired many future scientists and engineers due to its safety, low cost, and global popularity. Most often, small model rockets are not equipped with any kind of on-board avionics. A most basic avionics package records the flight parameters, providing data for determining values such as apogee or velocity. To add even the most basic avionics payload to a small model rocket, model rocketeers are often faced with the choice to either design it from scratch, or purchase an expensive proprietary module. As high-performance electronics have become smaller and more affordable, it is time for this to change.
This poster showcases the experience and lessons learned through prototyping, building, and testing a simple, yet versatile small-factor avionics module for model rockets. Current module capabilities include: logging flight parameters, live telemetry transmission, remote configuration, second stage firing, and easy extensibility in both hardware and software. The design is based on commonly available parts and can thus be built without a custom-manufactured PCB. The authors propose a vision for the future evolution of model rocket avionics and invite enthusiasts from all over the world to join in their project.

Posters and Demos / **43**

Current and Future use of Open Source in Space Science and Data Processing

Author: Tanuj Kumar[1]

[1] *Department of Electrical & Electronics Engineering, BITS, Pilani*

Corresponding Author: kmrtnjscnc@gmail.com

Open Source is still a relatively new term for many academics. They often do not know of open source works that are as useful as more widely used proprietary products. Reasons for this include lesser awareness of open source and its advantages, difficulty in migrating from existing proprietary systems, and sometimes, lack of documentation.

The first section of the paper will introduce the reader to low open source usage, not just in cubesat development, but in all of space science. Possible reasons for this, like an apparent inherent difficulty in its use will be discussed, thereby setting a direction to discuss possible solutions in the following sections.

The next section addresses these reasons by providing a comprehensive resource for space scientists to easily choose suitable open source works. This shall be discussed in two sub-sections: (i) Tools required in development of projects, and (ii) Tools required for data processing.
For example, the developmental phase may require software for simulations (Eg: GNU Octave, LT-Spice) or hardware for testing circuits (Eg: Arduino). The data from missions needs to be acquired (Eg: GNSS-SDR), and processed (Eg: OpenCV for image processing).

The paper will then discuss open data, a term that's infrequently heard of in the context of space science. Specifically, the growing need for open data, challenges posed by international and economic policies, ways to overcome them, and how a path to this concept is already being opened up (Eg: SpaceX) will be discussed.

Finally, the paper will discuss how people in Space Science can themselves contribute easily to this indispensable and ever growing resource, and conclude with a proposed vision for a complete, Open Space Science.

Posters and Demos / **46**

SDR Makerspace

Authors: Manolis Surligas[1]; Manthos Papamtthaiou[2]; Pierros Papadeas[None]

[1] *Libre Space Foundation, Computer Science Department University of Crete*

[2] *Libre Space Foundation*

Corresponding Author: surligas@gmail.com

Libre Space Foundation is cooperating with European Space Agency to exploit the advantages of Software Defined Radio (SDR) technology in space applications under the program codenamed "SDR Makerspace".

The goal of "SDR Makerspace" is to perform research and experiments with SDR software and hardware that will provide insights about the advantages or possible limitations of the technology for use in space applications. The activities of the programme, include experimentation with state of the art hardware, radiation testing of several SDR devices and their immunity on the harsh space conditions or development of new communication techniques. Moreover, the "SDR Makerspace", will investigate the reconfiguration capabilities of SDR equipped spacecrafts. We believe that such telecommunication schemes will play a key role in the near future, especially for small spacecrafts like Cubesats.

Continuing on the Libre Space Foundation philosophy and commitment on open source, all of the resulting code or hardware during the timeline of the activity will be publicly available. In addition, Libre Space Foundation will continue the community building activities focusing specifically on SDR-based communications, and utilizing its network to ensure synergy and upleveled awareness within the target community.

Posters and Demos / 48

An interdisciplinary CubeSAT program at HKU

Author: Pablo Saz Parkinson[1]

[1] *Physics Department & Laboratory for Space Research, The University of Hong Kong*

Corresponding Author: sazpark2@gmail.com

The Laboratory for Space Research (LSR) at The University of Hong Kong (HKU) is an interdisciplinary unit within the Faculty of Science, established in January 2016 and currently comprising approximately 25 members spread out over the Departments of Physics and Earth Sciences focusing on a broad range of research programs in high energy astrophysics, planetary science (including remote sensing) and space science. I will describe our plans to set up an HKU-led CubeSat program, building on our strong ties with various leading Chinese institutions and industry partners.