A COMPARATIVE STUDY OF UNIVERSITY SATELLITES ON-BOARD DATA HANDLING SOFTWARE PROJECTS

Waldo Acioli Falcão de Alencar, waldo@ita.br
Paulo Claudino Véras, pcv@ita.br
EmiliaVillani, evillani@ita.br
Departamento de Engenharia Mecânica, sala 253 – Instituto Tecnológico de Aeronáutica (ITA)
Praça Marechal Eduardo Gomes, 50 – Vila das Acácias, CEP 12.228-900 – São José dos Campos – SP – Brasil.

Abstract. *This paper presents a study with university satellite software projects and proposes a set of good practices for the development of On-board Data Handling embedded software. Considering high costs and complexity of commercial satellite projects, the characteristics of university satellite projects demand a different approach to systems engineering techniques than what is commonly used in industry. In order to have successful projects, these developments ask for robust, functional and well-defined devices. Educational institutions responsible for human resources in aerospace aim to simplify their satellite projects to attend all proposed requirements, considering reduced budget, a scarcity of manpower, and resources and focusing in professional training. Despite the quite large number of existing university satellite projects, there is still no construction template set for these products, including software implementation. Furthermore, different telemetry, telecommand, and data handling standards have been used. Considering the presented limitations about university satellites, the execution of the embedded software procedures has to be feasible. This is illustrated in comparison with the implementation of Brazilian Itasat satellite, in development by the Aeronautical Institute of Technology – ITA – where the procedures are ongoing. A survey was done using documents and published papers from 15 satellites, including ITASAT project, launched or in engineering model. The main goals identified in university satellites projects were the testing experiments and components not space qualified. These satellites operate with Real-Time Operating Systems, open-source or specially developed for the device, and communicate with their payloads through different protocols. A major number of launched equipments adopts packet standards for the management, engineering and product assurance in space projects and applications. Despite being simplified satellites, the software projects include error detection routines. Methods and techniques to support the assessment of software dependability and safety were often implemented. To highlight the results, a comparative summary is presented. Moreover, this work reinforces the importance of space technology professional training and the creation of a steady flow of engineers and scientists to work with high-level aerospace projects.*

Keywords: *university satellite, On-board data handling, embedded software*

1. INTRODUCTION

This paper discusses the development of data handling embedded software for university satellites, i.e., satellites that are developed by students in universities or educational institutions.

A good example of university satellite program is the CubeSat initiative (Heidt *et al*., 2000), satellites with the volume of 1 liter (10 cm^3). These satellites have low production cost. They are a good option for universities and private groups, such as amateur radio clubs, having access to space. Universities and educational institutions in aerospace area need to simplify their satellite projects to attend all proposed requirements, considering reduced budget, a scarcity of manpower and resources, and a focus in professional training. According to Thyagarajan *et al.* (2005), satellite development at the educational environment provides students with experience in satellite technology, from design through launch and orbit operations. This type of project requires integrated teamwork and provides valuable contribution to aerospace field, not only by training scientists and engineers but also as a tool for development of new technologies.

The context of university satellite programs differs from that of commercial or scientific satellite programs particularly for what concerns costs, available resources, complexity of the product and reliability requirements. Commercial or scientific satellites usually follow standards provided by agencies and organizations such as NASA and ECSS (European Cooperation for Space Standardization).

Theses standards contain detailed guidelines for space product development, which include the elaboration of extensive documentation, the following of costly manufacturing and testing procedures, the selection of radiation-hard components, among others. These guidelines aim at assuring the reliability of the space product. However, they augment the complexity of the project, as well as the necessary financial and human resource. As a consequence, university satellite programs demand an adapted development approach in order to be feasible.

Besides the great amount of university satellite programs, there is no standard or set of guidelines customized for the development of university satellites. In this context, this work analyses 38 university satellites projects, launched or stopped in the engineering model. The purpose is to identify common practices for the embedded software and the definition of telemetry, telecommand and data handling functions. The survey is based on published documents and papers. The results of the survey are discussed and compared with the solutions adopted by the Brazilian ITASAT satellite, under development at the Aeronautical Institute of Technology – ITA in partnership with other Brazilian universities and the National Institute for Space Research - INPE.

This works is organized as follows. Section 2 introduces the subsystems of a satellite and details the functions of the on board data handling subsystem, which is the focus of this work. Section 3 presents the ITASAT project and the solution adopted by ITASAT for the on board data handling subsystem. Section 4 presents the results of this survey and proposes good practices on low-cost satellite's software development. Section 5 brings some conclusions and discusses future work.

2. THE SATELLITE SUBSYSTEMS

A satellite is composed by the following subsystem, among others:

- Structure subsystem: consists of the physical structure that holds and protect all the satellite equipment;
- Power supply subsystem: generates and distributes the power for all the satellite equipment;
- Attitude and orbit control (AOCS): contains the sensors, actuators and control algorithms necessary for guiding the satellite to the desired orbit and attitude;
- TM/TC (telecommand and telemetry) subsystem: is responsible for the communication with the ground station;
- On-board data handling (OBDH) computer: process the telecommands and generates the telemetries;
- Thermal control: is responsible for assuring that all equipment stays in the appropriate range of temperature.

2.1. The OBDH Subsystem

The object of this paper is the OBDH subsystem. It is responsible for the internal communication of the satellite and data handling processes. According to Larson and Wertz work (1999), the OBDH subsystem performs two major functions: to receive, validate, decode and distribute commands to other satellite subsystems and payloads; and to gather, process, and format spacecraft housekeeping and mission data. This equipment often includes additional functions, such as timekeeping, watchdog monitor, and security interfaces. Common requirements of an OBDH are: high reliability, capability of real time processing, resistance to radiation, low power consumption, and reduced mass and volume (Vinci and Saotome, 2010). The OBDH subsystem's size is directly proportional to spacecraft complexity. If a spacecraft has a large number of subsystems and payloads, it requires more monitoring and configuration capability (Larson and Wertz, 1999).

A possible solution for the implementation of the OBDH subsystem is to share the same hardware with the AOCS. In this case, the on board computer runs both the OBDH software and the AOCS software tasks, which may be managed by an operating system.

The use of a computer to centralize all on-board processing functions is discussed by Jonas *et al.* (2002). The authors affirm that a satellite can be designed with intelligent subsystems, enabling them to accomplish their tasks and communicate without a distinct central computer. This would imply that all parts would need to have some sort of circuitry to perform intelligent decisions, for example microcontrollers. This solution would be very costly in terms of developing time, power dissipation, and space compared to a solution with only one centralized and more powerful computer. Furthermore, internal communication could turn out to be problematic with numerous microcontrollers. One disadvantage of this approach is that the all the satellite rely heavily on the on board computer. A fatal error in the computer will mean the lost of the mission.

In order to minimize this problem, redundancy should be used so that the on-board processors are replicated to increase the chance of success of the mission (Del Corso *et al.*, 2007). According to Vinci and Saotome (2010), other techniques that could also collaborate with the increase of on-board computer reliability are software redundancy, time redundancy, repeating computer processes that do not depend on the time.

2.2. Telemetry, telecommand and data handling standards

Usually, different telemetry, telecommand, and data handling standards are used for different satellites. Some agencies are working on standards for data handling systems. The Consultative Committee for Space Data Systems (CCSDS) is one of the agencies working on standardization of space/ground links, increasing the interoperability of their spacecraft and communications systems. As said in Koekemoer *et al.* work (1999), this committee is interested in developing standard data handling techniques to support space research for peaceful purposes, such as recommendations for telemetry and telecommand. Many of these recommendations serve as guide for other agencies standards, such as ECSS.

Rutter *et al.* (2001) consider the complete CCSDS TM and TC implementation still very complex for low-cost small satellites. For the referred author, the use of a cost effective and flexible communication system that consists of a simplified, yet reliable, software implementation of the CCSDS protocol is a good alternative solution. The software CCSDS implementation imposes minimal memory footprint and performance requirements on the on-board computer.

The reusability of systems for missions that adhere to the same data system standard provides the potential for large cost and reduced operations risks. However, these advantages do not apply to university satellites, where development teams usually change from one project to another.

Telemetry and telecommand subsystems on microsatellites have a history of technological advancement of over a half of century. Today these systems are designed to be very reliable, secure and redundant (Koekemoer *et al.*, 1999). The Packet Utilization Standard – PUS (ECSS, 2003) is part of ECSS standards and addresses the utilization of telecommand packets and telemetry source packets for the purposes of remote monitoring and control of subsystems and payloads.

Some missions implement a centralized architecture with a small number of application processes, while others have a highly-distributed architecture within which a correspondingly larger number of application processes are distributed across several on-board processors (ECSS, 2003). The PUS specification of services PUS is adapted to the expectation that different missions require different levels of complexity and functionality from a given service. So, the standard presents, for each service, a list of minimum capability set of subtypes for each type of service. The minimum capability set corresponds to the simplest possible level. This set allows the user to implement services of several distinct levels.

Figure 1 presents the frame structure of a telecommand or telemetry based on PUS. Each field is explained in detail in this ECSS standard (2003).

Packet Header (48 Bits)							Packet Data Field (Variable)			
Packet ID				Packet Sequence Control		Packet Length	D. Field Header	App. Data	Spare	P. Error Control
Version Number	Type	Data Field Header Flag	App. Process ID - APID	Seq. Flag	Seq. Count					
3	1	1	11	2	14					
16				16		16	Var.	Var.	Var.	16

Obs: numbers with this background color are in binary

CCSDS Secondary Header Flag	TC Packet PUS Version Number	Ack	Service Type	Service Subtype	Source ID	Spare
1	3	4	8	8	n	n

Figure 1. Generic telecommand or telemetry frame structure using PUS/CCSDS. Adapted from ECSS (2003).

Another used standard is the ESA PSS-45 (1978a) and ESA PSS-46 (1978b). This standard, old but simple to apply, defines the Pulse-Coded Module (PCM) telecommand (1978a) and PCM telemetry (1978b). Its simple frame structure is attractive for simple TM/TC systems. All the fields of the telecommand are repeated to ensure data reliability. Figure 2 presents a generic frame structure of a telecommand based on ESA PSS-45. The rules for telemetry frame structure are shown in chapter 3 of PCM Telemetry Standard (ESA PSS-46, 1978b).

1st Field	2nd Field	3rd Field	4th Field	5th Field	6th Field	7th Field	8th Field	9th Field
Sync.	Mode	Mode Repeated	1st Data	1st Data Repeated	2nd Data	2nd Data Repeated	3rd Data	3rd Data Repeated
16 bits	4 bits	4 bits	12 bits	12 bits	12 bits	12 bits	12 bits	12 bits

Figure 2. Generic PCM telecommand frame structure using PSS standard. Adapted from ESA (1978a).

Adaptations of these standards are also common in simplified designs of satellites - see Kasser (1992). These simplifications are usually made in satellites that were not designed with focus on system reliability, such as the ones from educational initiatives.

3. ITASAT PROJECT

The ITASAT Project is an effort to develop a university satellite within the scope of the Small Technological Satellite Development Program, funded by Brazilian Space Agency (AEB) with technical coordination of INPE and academic coordination of ITA (Yamaguti *et al.*, 2009). The efforts from AEB, INPE and ITA are to improve, develop and validate space technologies as well as to prepare and qualify human resources for the National Program of Space Activities (Yamaguti *et al.*, 2009). Recently, Pacheco *et al.* (2002) made a commentary about Brazilian space projects:

> "Brazil has a history of intense activity in the development of space technology, including small satellite construction infrastructure as well as good space qualifying and testing equipment and facilities. It has also developed different components to be

assembled into the International Space Station, along with multiple projects with foreign institutions and international organizations. Along with the Brazilian Space Agency, Brazil also relies on several national institutions and industry, which collaborate on the development of Brazilian space technology, chief among them the National Institute of Space Research (INPE in Portuguese)."

The university satellite ITASAT is a technological satellite with a payload for the continuity of the Brazilian Environmental Data Collection System. It will also carry some selected experiments as payloads. Figure 3 shows a layout of the ITASAT.

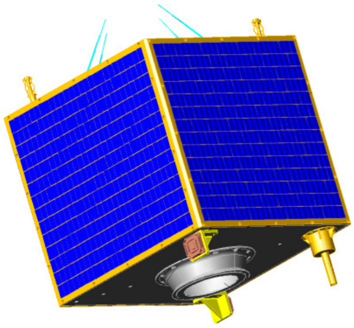

Figure 3. ITASAT Layout.

The ITASAT Mission composes the development, launch and operation of a small university technological satellite for use in a Low Earth Orbit (LEO). The satellite shall be capable of providing data collection services as offered by the Brazilian Environmental Data Collection System, besides offering a mean to test in orbit several experimental payloads (Yamaguti *et al.*, 2009).

One of the innovations of this aerospace platform is the development of an on-board computer that performs both the data handling functions (such as housekeeping) and attitude control algorithm. In this project, this configuration is called ACDH (Attitude Control and Data Handling). As a result of this integration, a high-performance embedded processor and reliable software shall be used. Table 1 describes some ITASAT general specifications. Table 2, adapted from Vinci and Saotome (2010), specifies some components used in the hardware centralized architecture of the on-board computer.

Table 1. ITASAT general specifications.

Attributes	**Satellite Specifications**
Size (cm^3)	60 x 60 x 60
Mass (kg)	80
Transceiver downlink (TM) data rate (kbps)	400 (nominal), 25 (safe)
Transceiver uplink (TC) data rate (kbps)	20
ACDH Processor	Atmel AT91RM9200
ACDH Power Consuption (W)	7
ACDH Memory	Start up Memory (EEPROM): ≥ 64 kbytes Program Memory (EEPROM): ≥ 4 Mbyte Execution Memory (SRAM): ≥ 8 Mbytes
On-Board Time Management (ms)	10

Table 2. ITASAT on-board computer hardware components. Adapted from Vinci and Saotome (2010).

Part Number	**Specified Components**
ARM9	AT91RM9200 ARM9 embedded processor
AT49BV040B	4 Megabit Flash Memory
K6R4008V1D	512K x 8-bit High Speed CMOS static RAM
RS-422	3V RS-422 DIFF DRV/RCV

The use of an ARM9 processor allows the use of a real-time operation system (RTOS) to coordinate all the tasks of the on board computer. The processing power of ARM9 presents no restriction barriers such execution time. In this project, RTEMS is chosen as the RTOS.

For the ground/spacecraft communication, the telecommand and telemetry packets are defined according to the standards ESA-PSS 45 (1978a) and ESA-PSS 46 (1978b). The data handling services are adapted from the PUS standard (ECSS, 2003), resulting in a combined solution that mix old and current standards.

As an aerospace platform with educational and technological development proposes, the university satellite is a good opportunity to evaluate the advantages of this mixture of standards.

The implemented services of PUS were:

- Service type 2: device command distribution service
- Service type 3: housekeeping & diagnostic data reporting service
- Service type 6: memory management service
- Service type 8: function management service
- Service type 9: time management service
- Service type 11: on-board operations scheduling service

Figures 4 and 5 illustrate how TM/TC will work on ITASAT.

1st field	2nd field	3rd field	4th field	5th field	6th field	7th field	8th field	9th field	10th field	11th field
Synchronisation	**TM Type**	**Digital Bi-level**	**ID Channel**	**Analog Single-Ended**	**Analog Double-Ended**	**Serial Digital 8-bits**	**Serial Digital 16-bits**	**Parallel Digital**	**Datation Ch. (Time-stamp)**	**Floating Format**
W0-W1	W2	W3-W7	W8	W9-W24	W25-W40	W41-W56	W57-W72	W73-W88	W89-W104	W105-W127
16 bits	8 bits	40 bits	8 bits	64 bits	64 bits	64 bits	64 bits	64 bits	64 bits	184 bits

Figure 4. Standard telemetry frame structure adapted from PSS standard.

1st field	2nd field	3rd field	4th field	5th field	6th field	7th field	8th field	9th field
S/C Address & Synchronisation	**Mode**	**Mode Repeated**	**1st Data**	**1st Data Repeated**	**2nd Data**	**2nd Data Repeated**	**3rd Data**	**3rd Data Repeated**
16 bits	4 bits	4 bits	12 bits	12 bits	12 bits	12 bits	12 bits	12 bits
96 bits								

Figure 5.Telecommand frame structure using PSS standard.

4. SURVEY OF UNIVERSITARY SATELLITE PROJECTS

In this work, a survey was done using documents and published papers from 38 satellites, identified as university projects, launched or in engineering model. The results are organized in a spreadsheet, to make them smoothly understandable. Figures 6 and 7 summarize the comparison.

	Satellite	University	Country	Year of	Primary OBC	Processor	Operating System (OS)	Secondary OBC	Memory
1	ITASAT	ITA	Brazil	2011	ACDH (ARM9+8051)	Atmel ARM9	RTEMS	Don't use	4Mb ROM, 8Mb RAM
2	KITSAT-1	Korea Advanced Institute of Science and Technology	South Korea	1994	OBC186 Mission Control Computer	Intel 80C186	SCOS	Z80-based MCC; T800 EIS and TMS320C30 DSPE cont.	Hybrid 13Mb
3	KITSAT-2	Korea Advanced Institute of Science and Technology	South Korea	1994	KASCOM (KAIST Satellite Computer)	Intel 80960 (32-bits)	KROS / SCOS	Don't use	Hybrid 13Mb
4	HUTSAT	Helsinki University of Technology	Finland	1996	HUTSAT OBC	Intel 80386SX	WOSNUC RTMTS	Duplicated OBC system	Not Specified
5	TSINGHUA-1	Tsinghua University	China	1999	Proprietary	Atmel ARM6 + GP2010 + GP2021	RTOS provided by the Mitel Semiconductors	8-bit microcontroller for TM/TC	Not Specified
6	DTUSat	Technical University of Denmark	Denmark	2002	Proprietary	Atmel ARM AT91M40800	eCOS	Don't use	PROM – min 8 KB; Flash – min 128 KB; SRAM – min 512
7	ChipSat	University of Surrey	UK	2001	Surrey Satellite Technology's (SSTL) SoC OBC	LEON 2-1.0.2a VHDL IP Sparc V8	VxWorks	Don't use	16MBits SRAM + 16MBits flash RAM
8	MEMSSat	Tsinghua University	China	2003	Proprietary	Atmel 32bits StrongARM 1110 4MRAM	Not Specified	Don't use	4 Mb RAM
9	JAESAT	Queensland University of Technology	Australia	2005	Intrinsyc CerfBoard	Intel XScale PXA255 microprocessor @ 400 MHz	Linux-based OS	Don't use	CompactFlash device
10	BAUMANETS	Bauman Moscow State Technical University	Russia	2006	Proprietary	Atmel ARM7DTMI processor @ 40 MHz	ECOS 2.0	Don't use	2MB PSRAM (pseudo static RAM), 2MB EPROM, 4MB
11	LEAP	National Cheng Kung University	Taiwan	2002	LEAP C&DH Diamond Prometheus	Not Specified	Not Specified	Don't use	2MB
12	FlyingLaptop-2	University of Stuttgart	Germany	2010	VIRTEX-II Pro FPGA XC2VP50	FPGA-based	Proprietary	FPGA (Spartan-II)	EEPROM, SSRAM, Flash
13	SUNSAT	Stellenbosch University	South Africa	1999	Proprietary 2-cores OBC	Not Specified	Not Specified	Don't use	RAM*
14	ASUSSat-1	Arizona State University	USA	2000	Proprietary	Intel 80C188 microprocessor	SCOS (BekTek Spacecraft Operating System)	Don't use	128k EPROM, 1M of RAM
15	PiCPoT	Politécnica di Torino	Italy	2007	Proprietary	Chipcon CC1010, and a Texas Instr. MSP430	Not Specified	Microchip PIC	MSP430 internal FLASH memory and FerroElectric

Figure 6.Spreadsheet overview, with the first information about the university satellites.

	Satellite	Comm between nodes	TM/TC Packet Standard	TM Transm. Rate	TC Transm. Rate	Language	Watch-dog?	COTS?
1	ITASAT	ISP, RS-422	ESA (PSS) + ECSS (PUS)	Data rate: 25 kbps	200 kbps (Nominal Mode), 25kbps (Safe Mode)	C	yes	yes
2	KITSAT-1	data sharing serial network	Not Specified	Not Specified	Not Specified	Not Specified	(Not Specified)	(Not Specified)
3	KITSAT-2	Serial / AX25	Not Specified	Not Specified	Not Specified	C	(Not Specified)	(Not Specified)
4	HUTSAT	Hybrid bus (parallel/serial interface)	Not Specified	38 kbps; 2.4 GHz for Hosekeeping	400 MHz for satellite control commands	Not Specified	yes	(Not Specified)
5	TSINGHUA-1	RS422 + CAN	Not Specified	Not Specified	N/S MHz for reset, power down, redundancy switches	Not Specified	(Not Specified)	(Not Specified)
6	DTUSat	Analog and Digital interfaces	Not Specified	Not Specified	Not Specified	C or ADA	yes	yes
7	ChipSat	Serial RS232, Parallel Port	CCSDS 102.0-B-4	Not Specified	(Not Specified)	C/C++	yes	yes
8	MEMSSat	RS232 Serial Bus	Not Specified	Not Specified	Not Specified	Not Specified	(Not Specified)	(Not Specified)
9	JAESAT	(Not Specified)	Not Specified	Not Specified	N/S MHz/9600 baud rate	Not Specified	(Not Specified)	yes
10	BAUMANETS	RS232, CAN2B interface, 16 input/output channels	Not Specified	Not Specified	Not Specified	C	(Not Specified)	yes
11	LEAP	Not Specified	Not Specified	430MHz with a date rate 1200bps	Not Specified	Not Specified	(Not Specified)	yes
12	FlyingLaptop-2	Not Specified	CCSDS 102.0-B-4	Not Specified	Not Specified	Handel-C	yes	yes
13	SUNSAT	CAN Protocol	CCSDS 102.0-B-4	8-bit Sampling at 1200 baud (0.39Hz) or 9600 baud	1200 baud (0.39Hz) or 9600 baud (3.125Hz)	Not Specified	(Not Specified)	(Not Specified)
14	ASUSSat-1	Serial RS232	AX.25	Not available	Not available	Not Specified	(Not Specified)	yes
15	PiCPoT	SPI	AX.25	FSK 437MHz	2440Mhz, 10kbps/s Baud Rate	C	yes	yes

Figure 7.Spreadsheet overview, with some software development information.

Research done in the documentation and published articles about university satellites shows several things in common between projects. All satellite projects have low cost attributes and carry experiments or subsystems to be tested. The main goals identified in a major number of university satellites projects are educational or human resources formation, and aerospace technology development, e.g. experiments and testing of components that are not space qualified.

Examples of projects can be cited. Stanford University established, in 1994, a university spacecraft project with the purpose of providing project based learning programs for engineering graduate and undergraduate students to gain experience in systems engineering. Additional goals for the university were to build the facilities, curriculum and research infrastructure for future laboratory programs (Heidt et al., 2000). At Universität Stuttgart, the "Flying Laptop" micro satellite is under development at their Institute of Space Systems. The primary mission objective is to demonstrate and qualify new small-satellite technologies for the future projects (Kuwahara *et al*., 2009).

Also, university satellites are often projected with a minor number of subsystems, simplifying their data handling system. According to the primary mission objectives from Flying Laptop-2 (Kuwahara *et al.,* 2009), the OBC itself will be the subject of evaluation for its innovative concepts and for qualification of its underlying hardware.

4.1. Features of the OBDH Software

Regarding the on board data handling software, several observations can be made about the survey's results.

Four from all analyzed projects use a secondary OBDH. The control of the Korean KITSAT-2(Kim et al., 1992) subsystems is performed by a multi-module on-board data handling. It consists of the primary OBC, TM/TC subsystem and the secondary OBC. The OBDH controls ADCS actuators, transfers software codes to payloads through data sharing network and dedicated parallel interface, stores payload results to its data storage for download, controls power module, exchanges data with ground station.

100% of the projects that specified their software development programming language used C or an adapted C language. For example, the KAISTSAT-4 (Korpela et al., 2003) flight software consists of 8500 lines of C code. Besides some routines that appear to be implemented in low-level languages, no reference to the use of assembler is made in any project papers.

Each satellite operates with a specific Real-Time Operating Systems. 9 of 11 identified RTOS are open-source or specially developed for the device. The exceptions were the Korean Kitsat-2 (Kim et al., 1992) and British CHIPsat (Janicik, Wolff, 2003). CHIPsat used a well-known commercial RTOS called VxWorks. For the Korean spacecraft, its primary OBC, a commercial real-time operating system, called SCOS, was purchased and used as its OS. The SCOS has been used on 80c186 because of its PC-based development environment and for providing priority based preemptive scheduling and intertask communication facility using message stream. When the RTOS is developed for the project, the use of a unique RTOS becomes a mission objective.

Despite being simplified satellites, the software projects include error detection routines. Methods and techniques to support the assessment of software dependability and safety were implemented. The Danish satellite DTUSAT (Jonas et al., 2002) implemented a software routine for error detection in the hardware. Even though the designers were very careful in the design process, physical manufacturing may lead to errors (bad soldering, short-circuits and others), which might be detected. The software helps in the debugging and test of the hardware. The KAISTSAT-4 OBC software implements detection and recovery from shutdown conditions. The recovery algorithms are active by default and can be overridden by ground command. Sunsat (Koekemoer et al., 1999) has little built-in fault tolerance on the T&T system, but the whole system is duplicated (dual redundancy), so a switch between systems can be done during system anomalies. British satellite CHIPsat implemented a system resettable without computer, where the reinitialization of the system may be achieved via a hardware command decoder which allows a back door reset of the system. Also, this system uses a global watchdog timer to, when necessary, automatically reset the computer. The university satellites ITASAT-1 and Flying Laptop-2 uses watchdog monitor and checksum comparisons. Watchdog appears to be an efficient status monitor, being evidenced in 6 projects.

An important aspect not evident in most projects is the possibility of upgrading the OBDH software. KAISTSAT-4 project (Korpela et al., 2003) highlights that its operating software can be uploaded in part or whole, in one or more orbital contacts. The code, resident in ROM, is RAM loaded at boot. If RAM load fails, or at the election of ground command, a ROM based uploader will be activated. Code is compiled in block boundaries with tags to facilitate modular upload modification of subroutines. ITASAT-1 is designed to be able to upload a new version in case of faults found in the current operating software. When the CPU module software receives a new program to replace the current program, the integrity of this new program received is verified through its checksum. The new program may be send as a telemetry frame to the ground station for confirmation of integrity. After the confirmation by the ground through telecommand that the new program uploaded is OK, the CPU module software shall wait for a telecommand to restart the OBC. The address of the new executable program shall be modified when requested by telecommand.

A major number of low cost satellites adopt standards for the management, engineering and product assurance in space projects and applications. Regarding the communication with the ground station, some standards are based on CCSDS recommendations or are not specified. Sunsat project (Koekemoer et al., 1999) adopts CCSDS recommendations related to packet telemetry and telecommand. Although these recommendations are generally employed on large spacecraft, it is possible to implement a subset on a microsatellite. The ChipSat project (Rutter et al., 2001) main requirements were to provide a simplification of the main CCSDS telemetry and telecommand protocol, avoiding long implementation time and cost, and ensuring reliable TC acknowledgments from the spacecraft. The development of the CCSDS software package is based on the CCSDS TM/TC Recommendation documents, which contain the detailed specifications of the logic required to achieve a CCSDS reliable communication system. ITASAT-1 uses ESA-PSS standards to define the telemetry and telecommand frame structure because of its simplicity. But also adapts that structure, using PUS services.

Kasser (1992) proposes a simple frame structure for binary telemetry in low cost satellites. The basic assumption is that the transfer frame is encapsulated in an AX.25 packet, containing a header, which identifies the source and destination (broadcast address) of the packet. The transfer frame standard inserts a secondary header into the data area prior to the data itself. This secondary header provides information about the data that may be used by ground station during an acquisition session and for telemetry logs. A similar approach is employed by the Italian PiCPoT spacecraft project. The KAISTSAT-4 project (Korpela et al., 2003) verifies each received command packet before execution and, at set intervals (which can be modified by command), the housekeeping monitors are both sent to the OBC and are stored into packets for transmission to the ground within the payload data.

4.2. Features of the OBDH Hardware

Satellites are normally extremely expensive and account for years of development. In order to improve the probability of success, it is common to choose components of great reliability for satellites. As pointed by Jonas et *al.* (2002), in order to assure reliability, it is common to buy components that have actually been tested in space. Another component just like it will probably work in space too. This approach is expensive. Moreover, as the development of satellites is a long process, when you find a component that has been successfully used in space, it is at least two or three years old (Jonas et *al*., 2002). Since a considerable number of university satellite on-board computers uses components off-the-shelf (COTS), it becomes the perfect place for experiments with new components, e.g., for testing that the components do not collapse when hit by space radiation. The CHIPSat system adopts a design based on COTS philosophy (Janicik, Wolff, 2003). 1000-hour burn-in testing were made with these components, like thermal stress screening, radiation testing, thermal cycling, and thermal vacuum. A positive result given from these devices allows them to be applicable to the space environment. The OBDH from University of Stuttgart’s satellite (Flying Laptop-2), also adopts a COTS approach and designed a highly redundant replicated four lane computer system.

Also, COTS interface technologies were used and seems to simplify some steps of the project, like design, build, and test of general satellite's communication bus. According to Janicik & Wolff (2003), for internal communications within the bus, for example, communication between OBDH and satellite's payloads, standard COTS interface protocols are used. Most common protocols observed in the survey's result were RS-422, RS-485, serial SPI and CAN protocol.

Major part of all projects choose 80x86 or ARM-based processors as primary on-board computer processor. Some projects use a secondary processor to perform specific tasks or works. Figure 8 illustrates, in a pie chart, the frequency of primary on-board computer processor in the analyzed projects.

Figure 8. Pie chart with the frequency of primary on-board computer processors used.

However, satellites like DTUSAT (Jonas *et al.*, 2002) and MEMSSAT (Jianping *et al.*, 2002) preferred to have a 32-bit processor, which will please the software groups very much, that uses less power than an older and less powerful 16-bit processor. The idea is to put less restriction in software group works.

4.3. Good Practices on Software Development for University Satellites

From the information collected by this survey, some points should be highlighted and considered as good practices in software development for low-cost satellites, especially satellites developed within a university environment.

Even if some satellites do not seem to use a real-time operating system, probably because of system's simplicity, 11 RTOS were identified in the surveys. For critical timed systems, the use of a real-time operating system is highly recommended.

Despite all the projects that have specified the programming language used to implement the software have adopted C / C + +, no formal conclusion can be drawn from this observation, but that this language is apparently common. No information about low-level languages was collected but they were probably forgotten or omitted from the papers.

Some care must be taken while developing critical software. Knowing that the system is not 100% reliable, error detection or fault tolerant routines must be implemented. The most commonly observed routines were:

- The recovery algorithms to detect and recover the system from shutdown conditions.
- System dual redundancy, enabling switch between systems during system anomalies.
- Watchdog timer to, when necessary, automatically reset the computer.
- Checksum comparisons, for detection of errors that may have been introduced during its transmission or storage.

Also, a routine of hardware error detection via software was observed. It might be implemented as an additional error checking in the hardware.

The possibility of a new OBDH software upload might be considered, even in low-cost projects. The upload of a new application software and aftermost restart of OBDH might save satellite in situations of deadlock and/or system crash.

Adopt aerospace standards to be used as main reference in aerospace software projects is recommended. Agreeing on standard requirements is fundamental to a successful project. CCSDS and ECSS standards were used in the studied projects. Although these recommendations are generally employed on large spacecraft, it is possible to implement a subset or a simplification of the main CCSDS telemetry and telecommand protocol, for example, for simpler satellites. Withal, a simple frame structure is proposed for binary telemetry, encapsulated in an AX.25 packet, widely used in amateur radio communication.

As 25% of the projects include a secondary module, the utilization of a auxiliary computer module may not be interpreted as necessary, but may help in some tasks such as housekeeping, checking telecommand packages and receiving direct telecommands.

5. CONCLUSIONS

This article presents a study about university satellite software projects. The proposal is to highlight common features of low-cost satellite software projects. The results have shown that low-cost spacecraft systems tend to adopt a number of low-cost strategies to avoid the use of radiation hard components and ensure via software and hardware the on board computer reliability. They fault detection routines are adapted to the need of each satellite.

A summary of the survey is presented and discussed in this paper. This paper also presents a proposal of good practices on software development for university satellites based on the observations made in the software projects analyzed.

Moreover, this work reinforces the importance of space technology professional training and the creation of a steady flow of engineers and scientists to work with high-level aerospace projects.

Possible directions for future research are: 1) the verification techniques used for on board computers of university satellites; 2) the use of modeling techniques in the software development of university satellite, and then; 3) a comparative study with low-cost satellite on-board computers, including hardware aspects, fault tolerance aspects, implementation, communication, tests and project verification & validation.

6. ACKNOWLEDGEMENTS

The author’s would like to thanks for the financial support of the Agência Espacial Brasileira – AEB and Instituto Nacional de Pesquisas Espaciais – INPE.

7. REFERENCES

Berget, Richard T., 1999. “Spacecraft Subsystems: Command and data handling.” In: LARSON, Wiley J.; WERTZ, James R. (Comp.). Space Mission Analysis and Design. 3. ed. El Segundo, Ca: Microcosm Press, 1999. Chapter 11, pp. 395-407.

Del Corso, D., Passerone, C., Reyneri, L.M., Sanso`e, C., Borri, M., Speretta, S. and Tranchero, M. 2007. “Architecture of a Small Low-Cost Satellite”. Proceedings of the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, Turin, Italy, pp. 428-431.

ESA – European Space Agency, 1978a. ESA PSS-45 (TTC.A.01): PCM Telecommand Standard. Issue No. 1.

ESA – European Space Agency, 1978b. ESA PSS-46 (TTC.A.02): PCM Telemetry Standard.

European Cooperation for Space Standardization, 2003. “ECSS-E-70-41A. Space engineering: Ground systems and operations – Telemetry and telecommand packet utilization”. ESA-ESTEC: ESA Publications Division, Noordwijk, The Netherlands, 228 p.

Gentina, J., Yamaguti, Y., Varotto, S.E.C., Waldmann, J., Fernandes, D., 2009. “A PROPOSAL FOR ITASAT SATELLITE CONFIGURATION AND ITS PRELIMINARY MISSION ANALYSIS”. Proceedings of Brazilian Symposium on Aerospace Eng. & Applications, 3rd CTA-DLR Workshop on Data Analysis & Flight Control, S. J. Campos, Brazil, 10 p.

Heidt H., Puig-Suari, J., Moore, A.S., Nakasuka, S., Twiggs, R.J. 2000. “CubeSat: A new Generation of Picosatellite for Education and Industry Low-Cost SpaceExperimentation”. Proceedings of 14th AIAA/USU Annual Conference on Small Satellites, Utah, U.S.A, pp 1-19.

INPE. 2009. “Satélite e seus subsistemas”, 1 Feb 2010, <http://www6.cptec.inpe.br/~grupoweb/Educacional/MACA_SSS/.>

Janicik, J., and Wolff, J., 2003. “The CHIPSat Spacecraft Design – Significant Science on a Low Budget”. UV/EUV and Visible Space Instrumentation for Astronomy II. Edited by Siegmund, Oswald H. W. Proceedings of the SPIE, Volume 5164, pp. 31-42

Jianping, X., Zhenyu, C., Zheng, Y., Ke, G., Huibo, J., 2002. “On-board computer Subsystem Design for the Tsinghua Nanosatellite”. Proceedings of the 20th AIAA International Communication Satellite Systems Conference and Exhibit. Montreal, Canada.

Jonas, S., Malte, B., Thomsen, M.B., 2002. “DTUSAT ON-BOARD COMPUTER FOR PICO SATELLITE”. Technical Relatory, Copenhagen, Denmark, p. 65.

Kasser, J.E., 1992. “AMATEUR SATELLITE TELEMETRY: PAST, PRESENT and FUTURE”. IEEE National Telesystems Conference, Washington D.C., USA., Vol. 12, pp. 09-17.

Kim, H.S., Park, J.H., Kim, K.H., Lee, H.K. and Choi, S.D., 1994. “ON-BOARD COMPUTER SYSTEM FOR KITSAT – THE FIRST KOREAN SATELLITE” Idenficar Revista ou Evento. Teajun, South Korea, pp. 34-39

Koekemoer, J.A., Bakkes, P. and Mostert, S., 1999. "DISTRIBUTED TELEMETRY & TELECOMMAND FOR FUTURE SUNSAT MISSIONS". Acta Astronautica Vol. 44, Nos. 7-12. Matieland, South Africa, pp. 531-541, 1999.

Korpela, E. J., Edelstein, J., Berg, P., Bowen, M. S., Chung, R., Feuerstein, M., Han, W., Hull, J. S., Jin, H., Lee, D.-H., Min, K.-W., Nam, U.-W., Nishikida, K., Rhee, J.-G., Ryu, K., Seon, K., Welsh, B. Y., & Yuk, I., 2003. “The SPEAR science payload” Proc. SPIE, Vol. 4854, Taejon, South Korea, p. 665.

Kuwahara, T., Böhringera F., Falke, A., Eickhoff, J., Huber, F. and Röser, H.P., 2009. "FPGA-based operational concept and payload data processing for the Flying Laptop satellite". Acta Astronautica, Vol. 65, Issues 11-12, Sttutgart, Germany, pp. 1616-1627

Pacheco, E., Conte, R., Mendieta, F.J., Muraoka, R., Medina, J.L., Arvizu, A., 2002. “The Development of a Micro-Satellite Platform for Space Technologies Knowledge and Human Resources Preparation”. Idenficar Revista ou Evento., Ensenada, Mexico, Vol.2, p. 667.

Rutter, I., Vladimirova, T., Tiggeler, H., 2001. “A CCSDS Software System for a Single-Chip On-board Computer of a Small Satellite”15th AIAA/Utah State University Conference on Small Satellites, Utah, USA, SSC01-VI-4.

Silva, E.C., Villani, E., 2009. “INTEGRANDO SysML E MODEL CHECKING PARA V&V DE SOFTWARE CRÍTICO ESPACIAL”. Proceedings of Brazilian Symposium on Aerospace Eng. & Applications, 3rd CTA-DLR Workshop on Data Analysis & Flight Control, S. J. Campos, Brazil.

Speretta, S., Reyneri, L., Sansoé, C., 2006. “PiCPoT’s telemetry streams”. Torino, Italy, p. 24. 10 feb. 2011. <polimage.polito.it/picpot/documentazione/PiCPoTs_telemetry.pdf>.

Thyagarajan, K.; Gupta, J.P.; Goel, P.S. Goel and Jayaraman K., 2005. “University small satellite program – ANUSAT”. Acta Astronautica, n. 56, Bangalore, India, pp.89-97. 27 nov. 2010 <www.elsevier.com/locate/actaastro>.

Vinci, E., Saotome, O., 2010. “Reliability of On-board Computer for ITASAT University Satellite”. 11th LatinAmerican Test Workshop (Proceedings). Punta Del Este, Uruguay.

Yamaguti,W., Carvalho, T.R. and Varotto, S. E. C., 2009. ”ITASAT Mission Requirements and System Definition”.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.