Aerospace Science and Technology 47 (2015) 388–395

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Hardware-in-the-loop environment for verification of a small satellite's on-board software

Michael Fritz [a], Sebastian Winter [a], Juergen Freund [a], Stefan Pflueger [a], Oliver Zeile [a], Jens Eickhoff [b], Hans-Peter Roeser [a]

[a] Institute of Space Systems, University of Stuttgart, Pfaffenwaldring 29, 70569 Stuttgart, Germany
[b] Airbus DS GmbH, Claude-Dornier-Straße, 88090 Immenstaad am Bodensee, Germany

ARTICLE INFO

Article history:
Received 27 December 2010
Received in revised form 21 April 2011
Accepted 18 September 2015
Available online 20 October 2015

Keywords:
System simulation
On-board software verification
Hardware-in-the-loop

ABSTRACT

Hardware-in-the-loop test beds are applied in industrial satellite development to verify on-board software and to simulate operational scenarios. Spacecraft attitude sensors are simulated by models, which eliminate the need to stimulate them for closed loop scenarios. Similarly, attitude actuators are simulated including actuation interaction with spacecraft dynamics behavior. This approach leads to an enormous reduction in a spacecraft's verification time. Small satellites developed by organizations like AMSAT or by universities are even more limited with respect to available staff and thus even more constrained. As a consequence, there is a need to implement a comparable approach for a hardware-in-the-loop simulation environment at lower cost than in industrial development. The University of Stuttgart has developed and verified a hardware-in-the-loop system test bed for a small satellite's on-board computer while maintaining compliance to industrial standards. Further cost reduction was achieved by applying an industrial system simulation kernel, additional open-source software, the voluntary participation of students, and cooperation with the industry. The system simulation environment connected to the on-board computer was verified by comparing its results with dedicated simulation tools of several disciplines. The latency of signal transfers between the on-board software and the simulator in closed loop operation was determined and analyzed with respect to compliance to control loops and simulation step sizes. All results have been proven to be satisfactory for application in university satellite development and are presented in this publication. This setup is now used in the frame of the on-board software verification for the Flying Laptop satellite.

1. Introduction

Small satellite development has become much more commercial and complex within the last decade. As a consequence, the test equipment must be accordingly state-of-the-art [1]. Industrial developers apply hardware-in-the-loop technologies for on-board software verification tests in order to reduce spacecraft development time [2]. The on-board computer is connected to a simulated satellite via its real interfaces applying real data protocols. This approach allows on-board software tests upon on-board computer breadboard models before the real spacecraft equipment hardware for platform, attitude control, and payload is available. Furthermore, there is no need for complex sensor stimulation setups to run realistic spacecraft operational scenarios because in such a setup everything except the on-board computer, including the rest of the spacecraft, its environment and dynamics, is simulated [3]. The University of Stuttgart has developed and verified such a hardware-in-the-loop environment which enables on-board software verification tests, based on the simulator infrastructure core developed by Astrium Satellites Friedrichshafen, to be performed. This setup is currently applied to the small satellite Flying Laptop (FLP) depicted in Fig. 1. The spacecraft has a mass of less than 120 kg and a size of $60 \times 70 \times 80$ cm^3 [4]. It applies to industrial command and control standards, e.g. the CCSDS telecommunication standard [5], the Packet Utilization Standard (PUS) recommended by the European Space Agency (ESA) [6] and the Remote Memory Access Protocol (RMAP) for SpaceWire interfaces [7]. This approach is an upcoming trend in recent small satellite development [8–10].

The simulation environment which allows simulations in real-time is based on a simulator kernel developed by Astrium Satellites. Thanks to cooperation with this company, the university was provided with this software [11]. The models of the satellite equip-

E-mail addresses: m.w.fritz@web.de (M. Fritz), jens.eickhoff@airbus.com (J. Eickhoff).

http://dx.doi.org/10.1016/j.ast.2015.09.020
1270-9638/

ment have been developed by students under the guidance of experienced Astrium engineers. Both the simulator and the on-board software are controlled by means of the ESA mission control system SCOS-2000. Automated tests can be performed with this system applying the procedure execution engine MOIS from Rhea System, another company cooperating with the Institute of Space Systems [12]. The applied software configuration is similar to setups used in industry, which gives students practical experience that can be applied to future employment opportunities [4]. Finally, during simulated flight scenarios the spacecraft can be visualized via the open-source software Celestia [13]. An outline of the entire setup is shown in Fig. 2.

2. On-board software verification methodology

Within the electrical architecture of the small satellite FLP, spacecraft equipment (e.g. reaction wheels) are connected to the On-Board Computer (OBC) via an I/O Board. The LEON3-based OBC core board interfaces with this I/O Board from the other side via SpaceWire (in fact two cold redundant OBC core boards are used). The spacecraft equipment, e.g. belonging to the attitude control system, the power supply system or the thermal subsystem, interfaces with the I/O Board according to their native interface types e.g. RS-422/485, I2C, TTL and others. This approach is depicted in Fig. 3. It is similar to industrial architectures as described in [10] and [14].

All of the connected equipment, including its functions, and the I/O Board is modelled in the simulator. This implies that the OBC core board hardware can be connected to the I/O Board model in the hardware-in-the-loop setup via an Ethernet–SpaceWire router, provided by 4Links Limited which again is a cooperation partner of the institute. Because the communication between OBC core hardware and I/O Board is based on the Remote Memory Access Protocol (RMAP), the I/O Board model must be able to handle this protocol and has to forward the included data via simulated native interfaces to the corresponding equipment models. These models must be capable of communicating not only via the real data protocols but also through simulated native interfaces as used in the hardware [15].

After connecting the OBC core board to the simulation environment, the on-board software can be verified from the bottom up during its verification campaign. It is necessary to define commands in the mission control system SCOS-2000 in order to control the on-board software. Furthermore, the simulated equipment models must provide the ability to simulate possible failures of equipment to perform failure detection and stress tests on the on-board software. Failure injection on simulation side can also be commanded from SCOS-2000 in the same test script. The on-board software test runs must cover all functions of all modules in the on-board software. Occurring problems in simulation runs are to be listed in a verification report which is generated semi-automatically from the simulator log files. This document contains feedback for the on-board software developers.

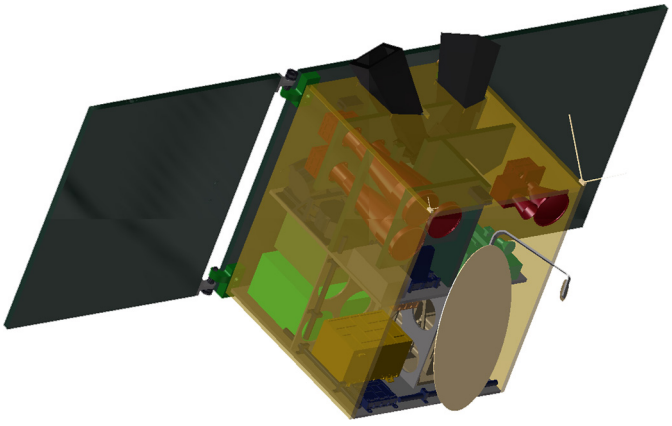

Fig. 1. Small satellite flying laptop. © Institute of Space Systems, University of Stuttgart.

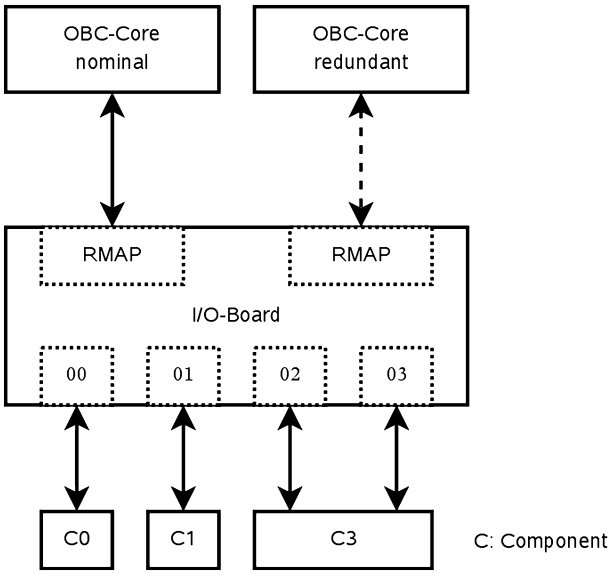

Fig. 3. Fundamental functionality of the Flying Laptop I/O boards [15].

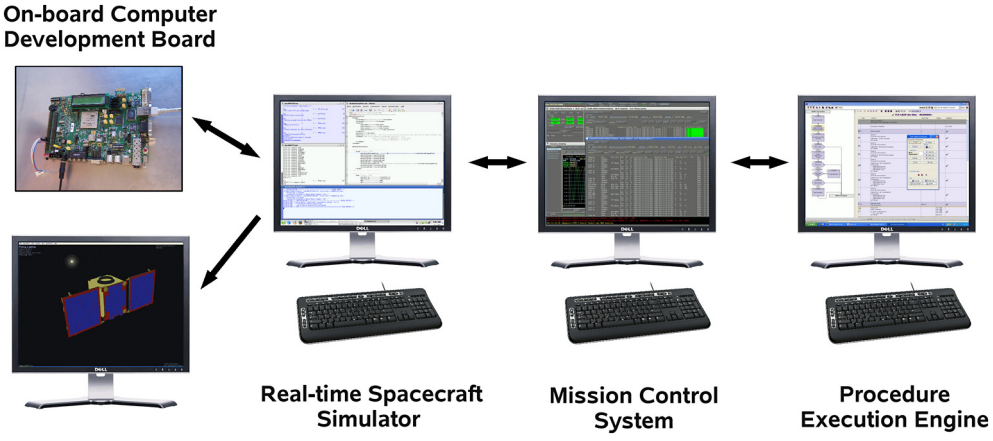

Fig. 2. General setup of simulation environment.