Modelling Interoperability between On-Board Computer and Payloads of the NanosatC-Br2 with Support of the UPPAAL Tool

Almeida, D. P. (1), Mattiello-Francisco, F. (2)
National Institute for Space Research – INPE
Av. dos Astronautas, 1758
São José dos Campos, Brazil
(1) danilopallamin@gmail.com
(2) fatima.mattiello@inpe.br

Abstract

This paper presents an experience in using a Model Driven Engineering (MDE) approach in the context of nanosatellite requirements verification. In the last ten years MDE approaches have been extensively used for the analysis of extra-functional properties of complex systems, like safety, dependability, security, predictability, quality of service. In order to anticipate the verification of the On-Board Software (OBSw) requirements in the NanosatC-Br2 development cycle, an MDE approach was used. Models of the interoperability between the on-board computer and the satellite's payloads were developed using *timed automata* theory with the support of UPPAAL, a tool environment for modelling, simulation and verification of real-time systems. From System and Interface Requirement Baseline documents, the models were developed focusing on interoperability requirements. The System Requirement Review (SRR), helped in reevaluating the requirements and raising new ones, leading to payload and OBSw operation redesign. The models have helped in the development of the OBSw and will provide possibilities of further development in testing and verification & validation with more exploration of the UPPAAL tool and its functionalities, such as the verification tool and automatic C code generation.

Keywords: CubeSat, NanosatC-Br2, Model-Driven Engineering, Timed Automata, UPPAAL

1- INTRODUCTION

Nanosatellite missions are commonly used in space-related technology development and testing, due to their low cost and short development cycle. They often use off-the-shelf components (COTS) and standardized subsystems to reduce the cost and development time. These characteristics also carry a downside: nanosatellite missions are generally much less tested than larger scaled and more expensive missions, and this leads to many cases of failures [1].

Universities and other academic institutions all around the world have groups with researchers and students designing, developing and testing nanosatellites and related systems. An example is NanosatC-Br2, the second nanosat of its family, a 2U CubeSat with scientific and technological purposes, being developed at Brazil's National Institute for Space Research (INPE) in partnership with groups from UFSM, UFRGS and UFMG [2]. It carries both software payloads running on the On-Board Computer (OBC) and physical payloads connected to the OBC via I2C bus.

This paper presents an experience in using a Model Driven Engineering (MDE) approach in the context of nanosatellite requirements verification, with the NanosatC-Br2 as a real application basis, in order to anticipate the On-board Software (OBSw) requirements. *Timed automata* models of the interoperability between the OBC and the payloads were developed using UPPAAL, a tool environment for modelling, simulation and verification of real-time systems. They were developed based on meetings with the developing teams, requirement baseline documents describing the payloads' operation, and system and interface requirements, and are still under refinement through an iterative process of model and payload updates.

The term interoperability describes the extent to which systems can operate simultaneously while exchanging data between each other.

The remainder of this paper is structured as follows. Section 2 presents the concept of MDE and UPPAAL tool. Section 3 describes the NanosatC-Br2's subsystems and the interfaces. Section 4 presents the modelling process, which is exemplified in section 5. Finally, section 6 presents the conclusion and future works.

2- MODEL DRIVEN ENGINEERING AND UPPAAL

Since the early days of computing, software developers have created abstractions to aid in software design. With the advance of more complex systems, graphical representations and models are being broadly used to better represent and help understand the systems. A Model Driven Engineering (MDE) approach to system design provides a high-level abstraction and the possibility to use graphical representations to help visualize the system, and also provides good tools for model-checking, that can detect many errors in early stages of the development life cycle.

This approach while using graphical representations, not only helps flatten the learning curve, but also permits that experts from other areas interact on the model to ensure the resultant software meets the requirements [3].

MDE aims to bring models to the center of the engineering and development process, having an important role in the specification, design, integration, validation and operation of systems. [4].

The tool used to develop the model, UPPAAL, is an environment for real-time system modelling, simulating and verifying. In this tool, the system is model as a network of *timed automata* (finite state machines extended with *clock* variables) operating in parallel. The clock variables in the *timed automata* are synchronized and evolve simultaneously, hence making these models appropriate for interoperability requirement verification.

UPPAAL is an open license platform for academic research and development. It has an internal simulator and model checker that can help deepen the model verification & validation (V&V) process [6]. Extension tools allow automatic C code generation for even further development and platform embedding for practical testing.

3- THE NANOSATC-BR2 PLATFORM

NanosatC-Br2 is a 2U platform that follows CubeSat standards. The main bus is entirely held in 1U, and all the physical payloads are allocated in the other U. It is an evolution of the NanosatC-Br1, INPE's first nanosatellite. NanosatC-Br2's mission is to validate and carry out scientific and technological experiments and develop a robust software for the on-board computer (OBSw). Along with this mission, INPE aims to build a solid basis for small satellite development and operation [2].

NanosatC-Br2 has 3 physical payloads:

- A Langmuir Probe being developed by a group in INPE. It will be used to measure electron density and kinetic temperature and also spectral distribution of plasma irregularities.
- A fault-tolerant attitude determination system being developed by a team of researchers at the Federal University of Minas Gerais (UFMG) and Federal University of the ABC (UFABC).
- A radiation-tolerant experiment composed of FPGAs and ASIC chips being developed by a team at the Federal University of Rio Grande do Sul (UFRGS) and Santa Maria Design House (SMDH).

The On-board Computer is connected to the payloads through the I2C bus, and following the I2C protocol, the OBC acts as the Master node and the payloads act as slaves.

Figure 1 represents NanosatC-Br2's subsystems and the interfaces among them. Depicted in yellow are the main bus's subsystems, in green is the OBC, in purple the daughterboard circuits and in blue the payloads.

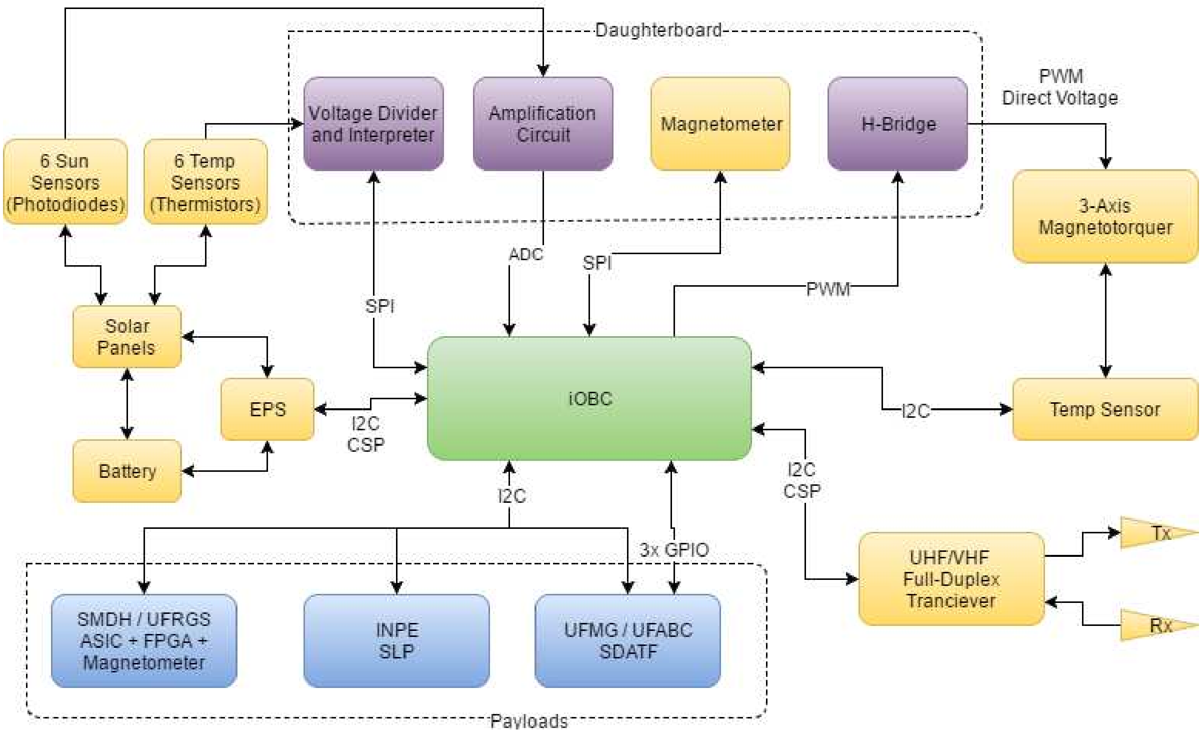

Figure 1: NanosatC-Br2 Subsystem Interface Representation

4- MODEL DEVELOPMENT

A model of interoperability between the OBC and each payload was developed by following each payload's operation requirement baseline documents and meetings with each developing team. These models simplify the systems to a high level of abstraction to represent only the states and transitions between states relevant to the interoperability between the OBC and the payloads.

With the development of each model, the requirements for the interoperability between the on-board computer and each payload were reviewed, refined and new requirements were added.

5- EXAMPLE MODEL

As an example of the models developed/in development, the model of the interoperability between the OBC and the SDATF payload is showed and described below:

As previously mentioned, the OBC acts as a Master to the SDATF Slave. The OBC has two major commands:

- **PCK_TASK**, where the OBC sends data from the last 1-10 readings of the sun sensors along with a timestamp of the readings, through a structured package. By sending this command, the OBC assumes that the payload will carry out the necessary calculations to determine the altitude based on the last altitude calculated and the new sun sensor readings, and shall wait the estimated time before sending other commands.

- **PCK_RESULT**, where the OBC requests telemetry of the last task sent. In case the OBC requests telemetry earlier than the specified calculation time or without previously sending a task, the payload will respond with an empty packet

The OBC must have the precaution of only saving sun sensor data when actually exposed to the Sun, as well as guaranteeing that the magnetorquers (attitude control actuators) are turned off (due to their interference with the readings of the magnetometer embedded in the payload).

The payload has basically one mode of operation, where it awaits the task command from the OBC while sampling the magnetometer. As soon as it receives the command it saves the readings of the magnetometer with the same timestamp as received with the sun sensors. Moreover, it uses these inputs along with the last attitude was calculated to determine the spacecraft's attitude.

In UPPAAL, states are represented by the gray circles, where the initial state has another circle inside. States have transitions between each other, which are represented by the arrows between them. The green writings are the *guards*, which are conditions for the transitions to happen. Dark blue means *updates*, which are actions taken when the transitions take place. In light blue are the *synchronizations*, which link the models (OBC with SDATF), where transitions that call the synchronizations end with "*!*" and transitions that listen for synchronizations (transitions that happen when the synchronization is called from another model) end with "*?*". In red are the state names and finally in pink are the *invariants*, which are conditions (commonly clock/timer conditions) for the model to stay in that state.

Figure 2 shows the OBC model of the interoperability, while Figure 3 shows the SDATF model:

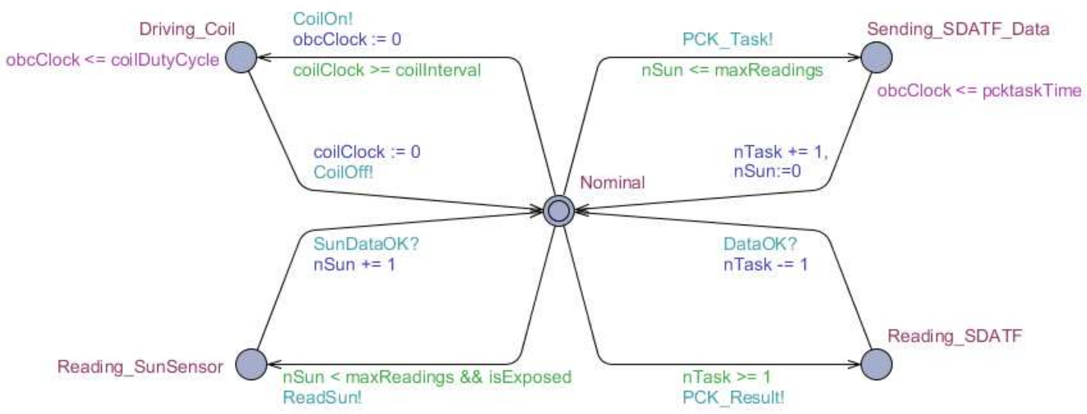

Figure 2: OBC Model for SDATF Interoperability

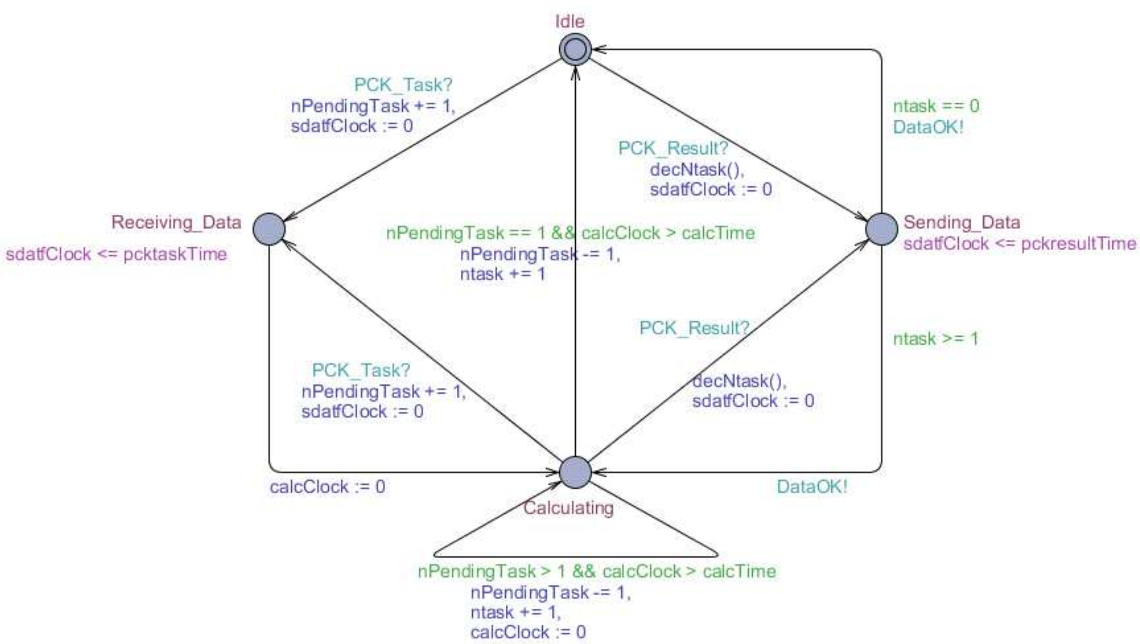

Figure 3: SDATF Interoperability Model with OBC

In the SDATF's operation, it starts calculating as soon as data is received. If the OBC sends data multiple times before requesting telemetry, the SDATF does the calculations for each package of data sent and stores the results in a FIFO buffer, so that the OBC can later request one by one.

Figure 4 shows, as example, part of the simulation cycle of the operation of both models:

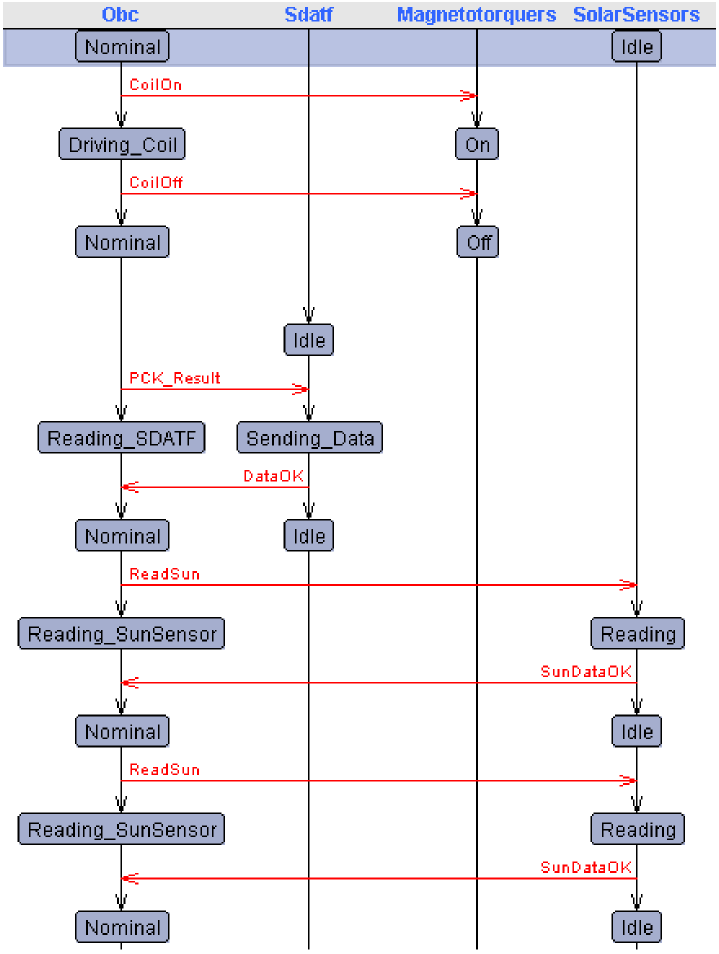

Figure 4: Example OBC and SDATF Model Simulation

It shows the interactions between the OBC and SDATF models, including the magnetorquers and solar sensors which have particular roles in the operation as described before. The interactions are represented as the red lines, and the model states are shown as the gray boxes. The simulation evolves vertically with each interaction. UPPAAL also provides in the GUI the values for the model variables updated on each step of the simulation.

6- CONCLUSION AND FUTURE WORK

The development of the UPPAAL models helped in evaluating and raising system requirements for the interoperability between the NanosatC-Br2's OBC and payloads. The high-level abstraction brought by using the MDE approach proved to be useful for requirement analysis. The models were used to discuss the system requirements in the System Requirement Review (SRR), where the modes of operation of the payloads were analyzed and lacks were identified, being redefined.

For future work, the models are being refined along with the evolution of the payloads. In addition, implementation of model-checking using UPPAAL is being studied for these models, along with C code generation using the UPPAAL2C tool extension for embedding the models in Arduino test setups [7]. These will complement the process of V&V being held on the platform.

7- REFERENCES

[1] Conceição, C. A. P. L, Ambrósio, A. M., Mattiello-Francisco, F.: On the use of NanosatC-Br Test System for payload operational requirements verification. IAA Conference Brazil 2016.

[2] NanosatC-Br2 Homepage. http://www.inpe.br/crs/nanosat/NanoSatCBR2.php. Accessed in March/17.

[3] Schmidt, D.: Model-Driven Engineering. Retrieved from https://www.fing.edu.uy/inco/grupos/coal/uploads/Main/mdepaper.pdf. Accessed in March/17.

[4] Montecchi, L., Lollini, P., Bondavalli, A.: An Intermediate Dependability Model for state-based dependability analysis. Techical Report, University of Firenze. Jan/2011.

[5] ESSADO, M. Documento de Requisito de Interfaces (IRD) – Versão 1.2. INPE / EMSISTI. Jul/2016.

[6] BEHRMANN, Gerd; DAVID, Alexandre; LARSEN, Kim G. A Tutorial on Uppaal 4.0. Aalborg University, Denmark. Novembro/2006

[7] Conceição, C. A. P. L., Batista, C. L. G., Mattiello-Francisco, F.: Dependability verification of nanosatellite embedded software supported by a reusable Test System. WDES 2016.