Institut für Software Engineering

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

# Design and Implementation of software tests for the ISS-experiment FARGO based on STPA-BDD

Michael Steinert

| | |
|---|---|
| **Studiengang:** | Softwaretechnik |
| **Prüfer/in:** | Prof. Dr. Stefan Wagner<br>PD Dr.-Ing. Georg Herdrich |
| **Betreuer/in:** | Manfred Ehresmann |
| **Beginn am:** | 01.04.2022 |
| **Beendet am:** | 30.09.2022 |
| **Berichtsnummer:** | IRS-22-S-043 |

## Abstract

STPA-BDD has been proposed for agile software development to facilitate the development of safety-critical software. This has already been tested in a controlled experiment, but beyond that insufficient real-world applications on the utilization of STPA-BDD have been published. To mitigate this issue and gain insights into the real-world utilization of STPA-BDD to enhance the process of developing safe software a specific use case is needed. Thus, in this thesis the development of respective software tests for the ISS-experiment FARGO is performed. A case study is conducted on the software test development part of FARGO, which uses the V-model as a working process. As expected STPA found additional failure cases to be considered. Analyzing the code coverage of the derived BDD scenarios required more effort than initially anticipated due to the fact that there was a discrepancy between the control structure used for STPA and the actual hardware. This was discovered when the BDD scenarios were about to be translated into test cases for the software. A solution to circumvent this issue was determined and implemented. It can be concluded that the assumption of STPA-BDD enhancing the development of safe software is technically correct, but further enhancement is possible and additional investigation is required.

## Kurzfassung

STPA-BDD wurde als Konzept für die agile Entwicklung von Software erarbeitet, um die Entwicklung sicherheitskritischer Software zu erleichtern. Dies wurde bereits in einem kontrollierten Experiment getestet, aber darüber hinaus wurde noch nicht genügend zur realen Anwendung von STPA-BDD veröffentlicht. Zur Minderung dieses Problems und um Einblicke in die reale Nutzung von STPA-BDD, das die Entwicklung sicherer Software erleichtern soll, zu gewinnen, ist ein spezifischer Anwendungsfall erforderlich. Daher wird in dieser Masterarbeit die Entwicklung entsprechender Softwaretests für das ISS-Experiment FARGO durchgeführt. Es wird eine Fallstudie am Softwaretestentwicklungsteil von FARGO, das das V-Modell als Arbeitsprozess verwendet, durchgeführt. Wie erwartet hat die STPA zusätzliche zu berücksitigende Fehlerfälle aufgedeckt. Die Analyse der Codeabdeckung der abgeleiteten BDD-Szenarien erforderte mehr Aufwand als ursprünglich erwartet. Der Grund hierfür ist eine Diskrepanz zwischen der Kontrollstruktur der STPA und der tatsächlichen Hardware. Dies wurde bei dem Versuch, die BDD-Szenarien in Testfälle für die Software umzuwandeln, entdeckt. Eine Lösung zur Umgehung dieses Problems wurde ermittelt und implementiert. Es kann geschlussfolgert werden, dass obige Annahme prinzipiell richtig ist, es jedoch Verbesserungspotential gibt und zusätzliche Untersuchungen erforderlich sind.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

FARGO (Ferrofluid Application Research Goes Orbital) is an experiment to be executed on a TangoLabs experiment rack in the microgravity ($\mu$G) environment of the ISS [1]. The experiment consists of three distinct experiments of electromagnetic ferrofluid manipulation as well as respective sensors to monitor experiment operation and results. Writing reliable, robust and safe-to-fly software is a non-trivial but necessary task for any a space application. The main aspect of completing this task successfully is testing the software. Behavior Driven Development (BDD) can help defining test cases to verify the software adequately [2, 3]. Prior to defining wanted behaviors the functional and performance requirements need to be known, especially the safety related ones. Special care for safety related behavior is required. To determine and verify all these requirements System-Theoretic Process Analysis (STPA) has been used successfully across diverse fields, like aerospace [4], automotive [5] and others [6, 7, 8, 9]. The combination of STPA and BDD for safety analysis and verification in agile development was investigated by Yang Wang and Stefan Wagner in 2018, showing promising results [10, 11, 12]. As a real world test the STPA-BDD concept is applied to define the software tests for the ISS-experiment FARGO, which is currently in development at the Institute of Space Systems (IRS) in cooperation with the Small Satellite Student Society at the University of Stuttgart (KSat e.V.) [13, 14, 15]. To get the testing as complete as possible we start by applying STPA to FARGO. With the results we define BDD scenarios along with respective tests.

In the Chapter 2 a short summary about STPA and BDD is given as well as a description of FARGO and its predecessor Pump Application using Pulsed Electromagnets for Liquid reLocation, the ISS-experiment PAPELL [16]. After that an overview where STPA has been used and extended is given in Chapter 3. A short explanation about STPA-BDD can also be found there. Chapter 4 shows how general conclusions about STPA-BDD can be made by applying it to a specific project. The STPA-BDD itself is split into three chapters, starting with the STPA in Chapter 5, going on with the BDD scenarios and tests in Chapter 6 and finally the validation of the software in Chapter 7. A conclusion is given in Chapter 8 and future work in Chapter 9.

# 2 Background

In this chapter brief background information about STPA, BDD, PAPELL and FARGO is provided.

## 2.1 STPA

System-Theoretic Process Analysis (STPA) is a technique for finding hazards and accidents imposed by a given system, based on Systems-Theoretic Accident Model and Processes (STAMP). STAMP and STPA were developed by Nancy Leveson at MIT to include formerly less focused on causal factors, e.g. organizational ones, flaws in design or software, component interaction, etc., into hazard analysis [17]. In STAMP accidents are seen as a control problem in which the controller of a process has a process model, which might be incorrect and can lead to unsafe control action to be executed on the controlled process. STPA is able to find these unsafe control actions in a design. To achieve this the accidents and hazards need to be identified and the control structure needs to be constructed as a prerequisite before the two steps of STPA can be performed. In the first step unsafe control actions are identified, in the second causal factors and control flaws. With this information safety constraints for the unsafe control actions can be deployed to mitigate the hazards [18].

## 2.2 BDD

Behavior-Driven Development (BDD) is a technique which encourages and strengthens the interaction between quality assessment and business analysis. It originated in agile software development with the goal to include all stakeholders into the development process. How the software should behave is specified in a structured way, but natural language is still used, so all stakeholders are able to understand what is written as specification for the software. The structure is:

- *Given* a context
- *If* a trigger
- *Then* expected outcome

Having one or more such scenarios for each use case of the software it can be implemented and tested against these scenarios and therefore proven to be correct [19].

## 2.3 PAPELL

Early in 2017 the first student competition for ISS experiments, called Überflieger, took place. The program was conducted by the German Aerospace Center (DLR) and the German Physical Societey (DPG), it allowed for three student teams to develop, build and operate their own microgravity experiments. “Pump Application using Pulsed Electromagnets for Liquid reLocation” (PAPELL), along with “Planet formation due to charge induced clustering on ISS” (ARISE) [20] and “Experimental Chondrule Formation at the ISS” (EXCISS) [21], was one of these three experiments [22]. With 15 months for a full project cycle the team behind PAPELL faced not only mechanical challenges [23], but also organizational ones [24]. In the end PAPELL was able to demonstrate a solid-state pumping mechanism, which is able to act as a digital microfluidic circuit [25], in microgravity ($\mu$G). It was developed by members of the Small Satellite Student Society (KSat e.V.) in cooperation with the Institute for Space Systems (IRS), both located at the University of Stuttgart, and carried out on the International Space Station (ISS) in 2018 [16]. The results of PAPELL lead to the “Ferrofluid Application Research Goes Orbital” (FARGO) [13] and the “Ferrofluid Application Study” (FerrAS) [26] projects, which aim to develop applications based on the shown capabilities of ferrofluids.

A description of the software of PAPELL can be found in the Student Experiment Document (SED) of PAPELL. The source code itself is in a repository on bitbucket [27]. The software of PAPELL was able to update the experiment configuration as well as itself. The chosen architecture for it is a finite state machine with four states, which can be seen in Figure 2.1. Note that only the software states for a nominal experiment run are shown, not the system states or non-nominal experiment runs. The system can be shutdown or go into safe mode at any time if needed. To enable different experiment runs on every boot a update mechanism is triggered. This checks if there is a new version of the software or new configuration for the experiment run and applys the changes. The update mechanism itself cannot be updated.

## 2.4 FARGO

FARGO, like PAPELL, will be operated on the ISS and investigate ferrofluid applications. The main difference is that this time the ferrofluids are incorporated in prototypes of applications, which aim to replace existing conventional mechanisms with low maintenance variants based on ferrofluids. In Figure 2.2 a rendered image of the current FARGO design iteration can be seen. The prototypes tested in FARGO are an attitude control system (ACS), an electrical switch and a thermal switch. The ACS actuator is operated like a Brush-Less Direct Current (BLDC) motor, thus it is sometimes referred to as ACS-BLDC. It improves upon existing flywheel-based ACS by replacing the bearing with ferrofluid. Both switches work by moving ferrofluid to enable or disable electrical current flow and respectively thermal flow. Moving the ferrofluid of the switches is achieved by switching an Electro Permanent Magnet (EPM) to minimize power consumption.

Significant data is to be gathered, processed and downlinked along with operating the experiment. To accomplish this two micro computers (Raspberry Pi Zero 2 W), a micro controller ($\mu$C, Raspberry Pi Pico) and an Electronic Speed Controller (ESC) are part of FARGO. The Pi Pico is the main On-Board Computer (OBC), being responsible for communication with the TangoLab and controlling both of the experiment computers. The control over the experiments goes as far as hard resetting

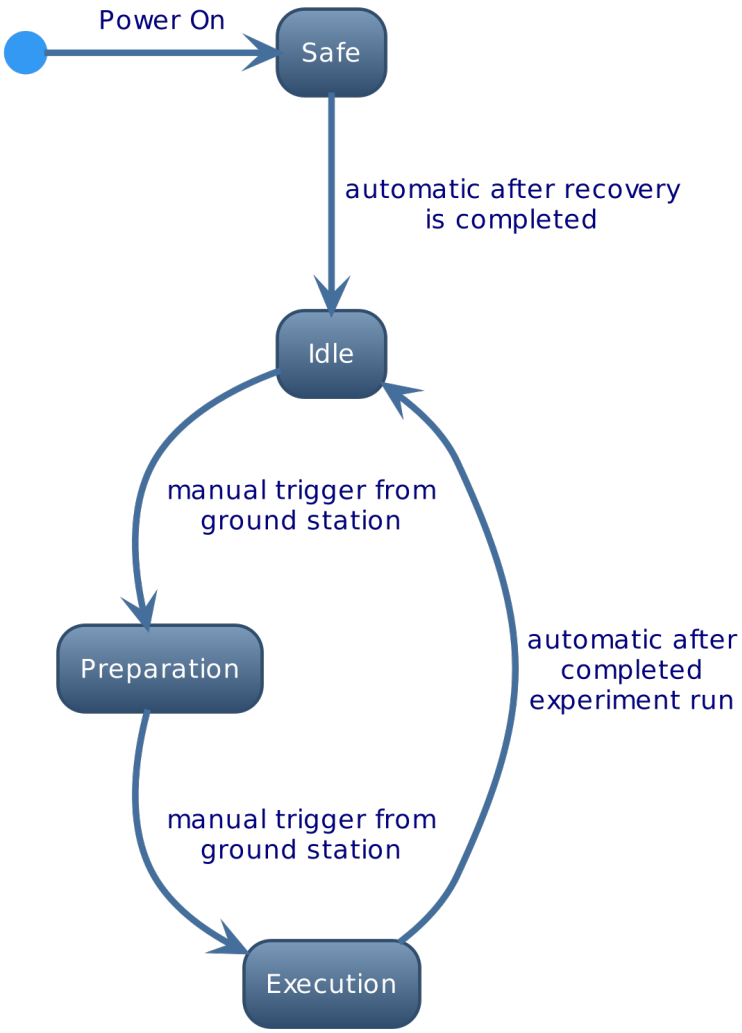


**Figure 2.1:** Finite state machine of PAPELL for a nominal experiment run

them by disabling their power supply. One of the Pi Zeros operates the ACS experiment, while the other one is responsible for the switch experiments. Thus they are referred to as “Pi Zero ACS” and “Pi Zero Switches” respectively. An overview of these components can be seen in Figure 2.3. External components are in grey, while programmable components are in green, actuators are in red and sensors are in purple.

The system is designed to be operated autonomously with no interaction from the crew on-board the ISS other than the integration of the TangoLab. While communication with the experiment in orbit is possible, it is limited to the options Space Tango provides, which is mainly the download of data, but also the possibility to send commands. This leads to several requirements regarding the software, e.g. high levels of reliability without limiting the operations team on what they can run experiment-wise. To fulfill these requirements the software is developed and tested using STPA-BDD, a combination of system theoretic process analysis and behavior driven development developed at the Institute of Software Engineering (ISTE), which is located at the University of Stuttgart.

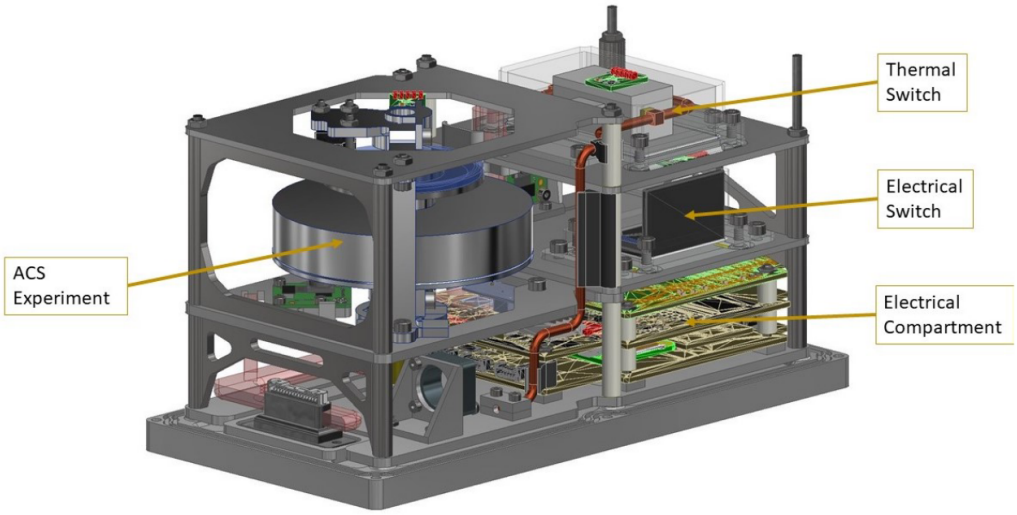


**Figure 2.2:** Rendered CAD image of the ISS-experiment FARGO internal setp, with the three experiments and the electrical compartment highlighted [14]

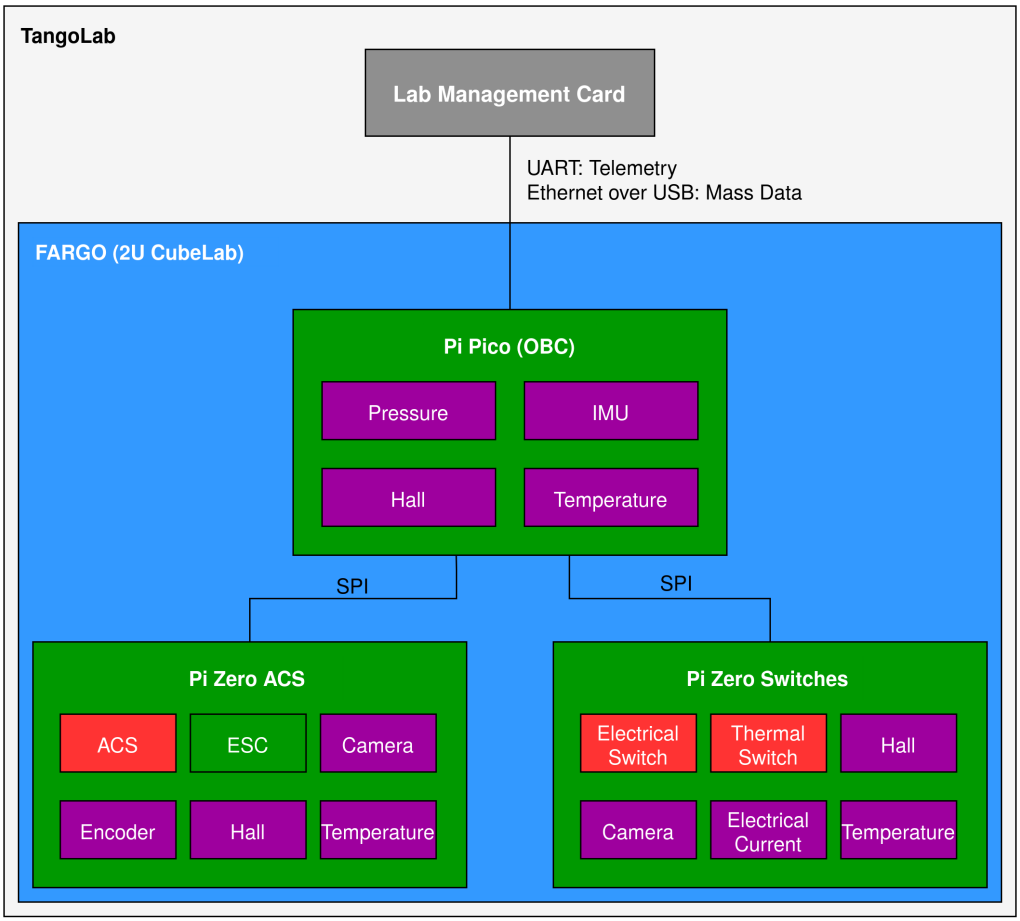


**Figure 2.3:** High level overview of the components of FARGO; grey: external components, green: programmable components, red: actuators, purple: sensors

# 3 Related Work

While STPA takes the software of a system into account like any other part, this does not necessarily mean that the software development process gets all the benefits it could get from the STPA approach. This is due to software engineers not being responsible for safety analysis and safety engineers not being responsible for software development. To resolve this issue Abdulkhaleq, Wagner and Leveson developed a safety engineering approach for software-intensive systems. This approach is based on STPA and enables taking safety into account in the software development process from the beginning. Safety requirements for the software are derived with STPA and then formalized. The formalization step enables the use of formal verification methods on the software and the generation of safety-based test cases [28]. STPA has also been extended to understand human behavior. A case study on automated parking systems was conducted therefore. The result was that with this extension it is possible to compare how different system designs affect human behavior [29]. Another example from the automotive domain is the integration of different assistance systems, where STPA is able to find conflicts between the systems, which otherwise would not have been found in the design phase [30]. In another case study the feasibility of using STPA in a system-of-systems was tested. The system-of-systems was an automated quarry site where the individual systems were already safety certified but the interaction of them not yet. It was found that STPA works for this kind of problem but is no perfect fit [31]. In the space domain STPA can help a lot with its ability to find issues early in the design phase [32]. It was found that most modern accidents originate not from component failure but from design errors [33]. The MOXIE-experiment received improved safety control loops due to the results of STPA conducted on it. While this was focused on the solid oxide electrolysis subsystem, it provided a framework for an expanded safety analysis [34]. Finding issues and taking care of them during the development is only the first step. Making sure that the issues are adequately considered is highly critical to mission success and is solved by qualification testing. To develop test cases for safety-critical systems several methods exist. Misuse cases, which are essentially use cases for breaking the system, can be used to get an insight on what failure cases are present. From there they can be used to develop safety requirements [35]. In case there are already successful test cases available they can be used to derive more test cases by using metamorphic testing. This can also be performed on systems that are already in production and test oracles are not required [36]. Another approach is to use scenarios to derive test cases. The idea of the SCENT method is to have an informal spec, which allows for more stakeholders to participate in the process of defining scenarios [37]. It is even possible to turn uncritical scenarios into critical ones by slightly modifying them [38]. STPA-BDD was developed as part of Safe Scrum (S-Scrum) by Yang Wang and Stefan Wagner at the Institute of Software Engineering at the University of Stuttgart. It starts with a STPA safety analysis conducted by a safety analyst [12]. The outcome of this analysis is a STPA safety report containing unsafe control actions, process variables, algorithms and other safety related concerns. This report is the input to the so called “3 Amigos Meeting”, a meeting between the safety analyst, the developer and the business analyst. In this meeting they define the safe scenarios and test cases for the BDD safety verification of the software. With this BDD scenarios the developer is able to write the software. Pending or failed test cases represent

unsafe scenarios, which are added to the STPA safety report for the next iteration. This procedure is visualized in Figure 3.1. In experiments STPA-BDD showed good communication effectiveness.

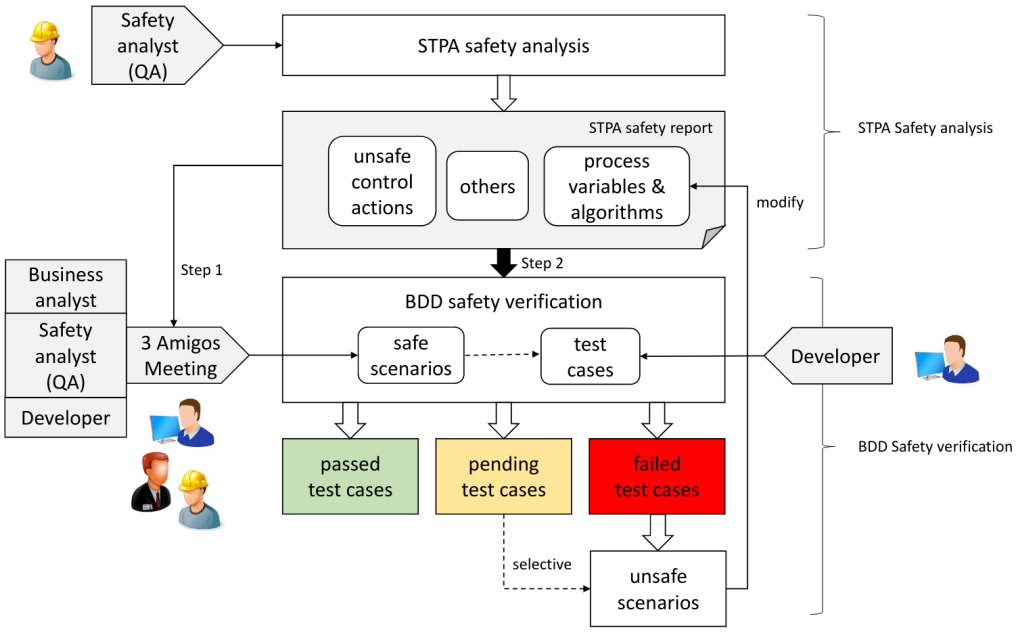


**Figure 3.1:** STPA-BDD Concept overview [12]

However fault detection effectiveness, productivity and test thoroughness showed no significant difference to user acceptance tests (UAT) until the BDD verification part was accelerated by the implementation of a semi-automated tool. With this tool all areas were improved by at least a factor of 1.5 [12, 11].

# 4 Research Method

As described in Chapter 3 STPA-BDD has already been tested in controlled experiments with Computer Science students [10]. The next step is to test it in a space project with Aerospace Engineering students, to see if the method is feasible in another domain. An important difference to the already conducted experiments is that the FARGO project is not using an agile process, instead it uses the V-Model [39].

To get a deeper understanding this work conducts a case study using the method defined by Yin. Briefly described the first step is defining a preliminary theory, then collecting data to check the preliminary theory and at the end deriving a final theory based on the collected data compared to the preliminary theory [40]. Due to STPA-BDD being a novel approach for developing safe software there is not much work on how good this approach is to be found. Applying the approach on a project is able to show the feasibility from this currently mostly theoretical approach in practice. Given these circumstances a case study is a productive approach to extend research on STPA-BDD.

The preliminary theory for this case study is that STPA-BDD is a ready to use tool to define software tests for safety-critical systems. To see if the STPA-BDD approach is an improvement to already in use approaches or just another way to define software tests the first question to answer is if the STPA part is capable to find issues that are not found by the traditional approach. The answer is given by conducting the STPA on FARGO and comparing its results to the verification matrix that has been defined for the critical design review (CDR), which was passed. In case the STPA part provides no improvement, the BDD part still could be of use as the BDD scenarios can be derived from the verification matrix as well. The next question therefore is which parts of the code are covered by the BDD scenarios. It is answered by translating the scenarios in actual test cases for the software. The working principle of STPA-BDD implies that the original UCAs are mitigated, but makes no assumption of introducing new UCAs via the mitigation. In S-Scrum, where STPA-BDD originates, this is not an issue due to the sprints of Scrum, thus there being STPA conducted multiple times. Whether this works for a non-agile process is also investigated before the gained insights are used to refine the preliminary theory.

Naturally this approach is not free of risks. It is to be noted that the participants of FARGO are aerospace students which have not finished their studies yet and are barely trained in software development. This should be not a real risk as one of the goals of this work is to find out if the approach can be used in a non-software domain. The existence of parts of the software is a real risk however. Having parts of the software in a working manner has a high chance of testing them less thoroughly by avoiding the hassle of putting them through the STPA-BDD approach or throwing them away and start from scratch instead.

# 5 STPA on FARGO

In this chapter a STPA is conducted on FARGO. It starts with the accidents and hazards sub-chapter in which in addition to its name the scope, system boundary and system-level constraints are defined. Then the components of FARGO are explained to enable a more straight forward understanding of the control structure, which is explained then. The control structure is needed to determine the UCAs and, at the end of the STPA, their causal factors. After that the research question "did the STPA find issues that would have been overlooked without it ?" is answered in the summary sub-chapter.

## 5.1 Accidents and Hazards

To be able to define the accidents and hazards at first the scope of the STPA needs to be defined. Given the nature of the Überflieger 2 program, which is being explicitly designed for student experiments [22], and the limitations enforced on the FARGO experiment it is hard for FARGO to impose a threat to astronauts. These limitations include requirements, which FARGO must fulfill in order to be launched, e.g. the outer shell must stay below 45°C at any time [41], as well as technically enforced ones, like maximum power draw. Damaging itself or tripping a fuse, which cannot be reset, and thus becoming a failure on the other hand is quite possible for FARGO. The scope of the STPA is therefore on safety in the sense of staying operational and experiment performance as well.

Another thing that needs to be defined before being able to define accidents and hazards is the boundary of the system the STPA looks at. This system boundary is required to distinguish between accidents and hazards, which can lead to accidents. For the STPA on FARGO the boundary will be the outer shell of the cube lab, the limits of the portal provided by Space Tango and the membership in the FARGO project. The boundary is defined this way to reflect the degree of control someone or something has from the perspective of the FARGO experiment. Naturally the project has some control over its members, especially over who will be able to operate the experiment in orbit and therefore having the chance to make a mistake. The outer shell boundary results from the FARGO team being responsible for everything inside it, while everything outside is not controllable by the team. The least amount of control is found in the portal, where the FARGO operations team members are just users that cannot change how the portal operatess.

With the scope and system boundary defined the accidents can be defined. For the FARGO experiment these are:

- L-1: (partial) mission loss
- L-2: violation of limitations defined by Space Tango

L-1 results from the desire to gain as much insight and data as possible. The reasoning for L-2 is that all requirements defined by Space Tango, especially safety related ones like maximal stored energy, must be satisfied to get qualification for launch and operation on the ISS.

After defining the accidents the hazards, which can lead to the defined accidents, can be listed:

- H-1: FARGO not able to accept commands [L-1]
- H-2: FARGO not able to collect data [L-1]
- H-3: FARGO not able to provide data [L-1]
- H-4: FARGO stores too much rotational energy [L-2]
- H-5: FARGO draws too much electrical power [L-1, L-2]

H-1 to H-3 are in the order of severity of the mission loss (L-1) they can cause, with H-1 being the most severe one. If FARGO is unable to accept commands (H-1) the mission is completely lost as there are no preprogrammed experiment runs planned yet. The inability to collect data (H-2) is nearly as severe as H-1. It is slightly less harmful because some endurance data along with environmental data, like power usage of the whole module, gathered and provided by Space Tango is still available. An example for endurance data would be the degradation of the ferrofluid and other data gained up on inspection of the returned cube lab. In hazard H-3 FARGO is working as expected, except for the provisioning of data over the downlink. This would lead to data being collected but not usable until after the return and disassembly of the cube lab, therefore the opportunity to tune subsequent experiment runs based on earlier ones is not available. Thus the collected data is less targeted compared to having the chance to use prior gathered data. Drawing too much power (H-5) is not only a problem with violation of limitations (L-2) but can trigger a fuse resulting in power to the cube lab being cut permanently, which would be a total mission loss (L-1). H-4 covers a hazard which could lead to L-2 but in this case the limitation cannot be enforced from outside the system as the information about how much rotational energy is stored only exists within it.

From the hazards the following system-level constraints are defined:

- SC-1: FARGO must be able to accept commands [H-1]
- SC-2: FARGO must be able to collect data [H-2]
- SC-3: FARGO must be able to provide data [H-3]
- SC-4: FARGO must not store too much rotational energy [H-4]
- SC-5: FARGO must not draw too much power [H-5]

The system-level constraints are the result of inverting the hazards. This is required for the STPA as the procedure is to find ways in which these can be violated. SC-1 and SC-3 could be summarized in a single constraint stating that FARGO must be able to communicate, as it is possible for constraints to cover more than one hazard. It was chosen to keep the constraints separated due to the difference in severity of the accident they can cause. Disobeying SC-3 would result in at least a partial mission loss, while underestemating SC-1 causes a total mission loss.

## 5.2 Components of FARGO

From a systems point of view the FARGO cube lab module can be divided in three main components: the main OBC, the ACS subsystem and the switches subsystem. This segmentation is also reflected in the electronics as each of the mentioned subsystems has its own printed circuit board (PCB). The boards are referred to as ACS board, switches board and main board respectively. On each of the PCBs there is at least one controller which needs to be programmed and interacts with sensors, actuators and controllers outside its board. The location of the PCBs within FARGO is the electrical compartment as shown in Figure 2.2. In the following sub-chapters a more detailed look is taken on each of these components.

### 5.2.1 Main OBC / Mainboard

The main OBC is a Raspberry Pi Pico, which sits on the main board. Its tasks involve gathering environmental or housekeeping data, relaying commands for the experiment runs to the respective subsystems, cache the results of experiment runs for downlinking to the ground station and monitoring the power draw of the ACS and switches. The housekeeping data gathered consists of the temperature, pressure and magnetic field within the cube lab. In addition the movement of FARGO is detected by an IMU. While the temperature and hall sensor, which detects the magnetic field, are expected to show changing values depending on the operation of the subsystems of FARGO, the pressure sensor and IMU should measure constant values in nominal operation. The only exception from this occurs in case the ISS is undergoing boosting or debris avoidance maneuvers, where the IMU will report non-constant values but the operation of FARGO still is nominal. There are four communication links in total, two to communicate internally and two for communication with the outside world. The subsystems are connected to the main OBC over SPI. The out-bound connection are specified by Space Tango as one UART connection for telemetry in a given format and one ethernet over USB connection for downloading arbitrary experiment data. Each of the three boards has its own 12V power supply line and each line is equipped with a power monitoring sensor read by the main OBC. The two power lines for the switch and ACS subsystems each have an electronic fuse in addition. These fuses can be triggered and reset by the main OBC to prevent triggering the permanent fuse for the whole experiment or to execute a hard reset if necessary.

### 5.2.2 ACS subsystem

On the ACS board is a Raspberry Pi Zero 2 single board computer (SBC) as main controller of the ACS subsystem. It is accompanied by a STEVAL-ESC001V1 (referred to as STEVAL board), which is a BLDC motor controller built around a STM32 ARM-processor. The task of the ACS subsystem is to evaluate a novel attitude control system based on ferrofluid. To achieve this the three phases of a modified BLDC motor need to be controlled and data about its temperature, magnetic field and rotation behavior needs to be collected. For outreach and public relations (PR) purposes in addition to the scientific ones the operating ACS is captured on video, which requires a light source inside the ACS compartment of FARGO. To get the modified BLDC motor the rotor of it is replaced with a custom one containing neodymium magnets and the bearings are replaced with ferrofluid. The three phases and thus the revolutions per minute (RPM), are controlled by the STEVAL board. The desired RPM count is communicated to the STEVAL board by the Pi Zero

via a pulse width modulated (PWM) signal. There is an UART connection between the STEVAL board and the SBC as well. It is used to get debug information and internal measurements from the BLDC controller and can also be used to update its configuration. Similarly to the main board a temperature and a hall sensor gather data. The difference is that the sensors are placed close to the coils of the BLDC motors stator to measure their temperature and magnetic field. Video capture is implemented with a Raspberry Pi Camera, referred to as Pi Cam, and illumination is provided by LEDs. Rotation behavior data is collected with two encoders, one with high resolution and one with low resolution. The decision for a second encoder with lower resolution but way more mechanical tolerance was made after assembling the high resolution one showed challenges in accuracy when manual integration is required. It is a fallback option as the high precision encoder might get outside of its mechanical tolerances during transport and therefore is unable to provide proper measurements, if any at all. A rendered image of the ACS experiment can be seen in Figure 5.1.

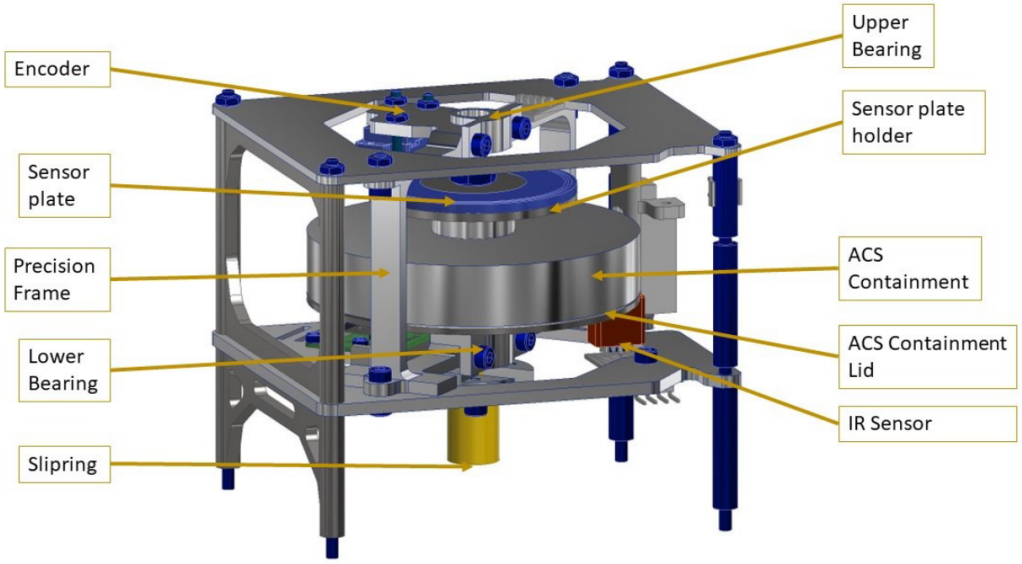


**Figure 5.1:** Rendered image of the ACS experiment with highlighted components [14]

### 5.2.3 Switches subsystem

The switches subsystem is controlled by a SBC like the ACS subsystem, but unlike has no additional controller to the Raspberry Pi Zero 2. Its task is conducting the experiments for the electrical and the thermal switch respectively. This includes operating the corresponding switch via an electro permanent magnet (EPM) and gathering data about them via various sensors. The electrical switch aims to replace traditional electrical switches with a mechanical free version that is not suspect to wear and tear. Like a traditional switch it has an on and an off state. The difference lies in replacing the contactor with a conductive ferrofluid that can be moved via magnetic fields. Behind a power monitor a capacitive, an inductive and a resistive load can be connected to measure the switches performance under different load scenarios. A hall sensor records the magnetic field in this experiment and a camera has the goal to capture the movement of the ferrofluid within the switch for the same purposes as the camera in the ACS experiment. A rendered image of the electrical switch

experiment can be seen in Figure 5.2. The technical functionality of the thermal switch is similar to the electrical switch, but instead of electrical current it switches thermal flow on and off. To achieve this a thermal conductive ferrofluid is moved between a heat source and a heat sink. Source and sink are peletier elements which in combination with multiple temperature sensors allow for precise temperature control and measurement. In contrast to the two other experiments this one has no camera. A rendered image of the electrical switch experiment can be seen in Figure 5.3.

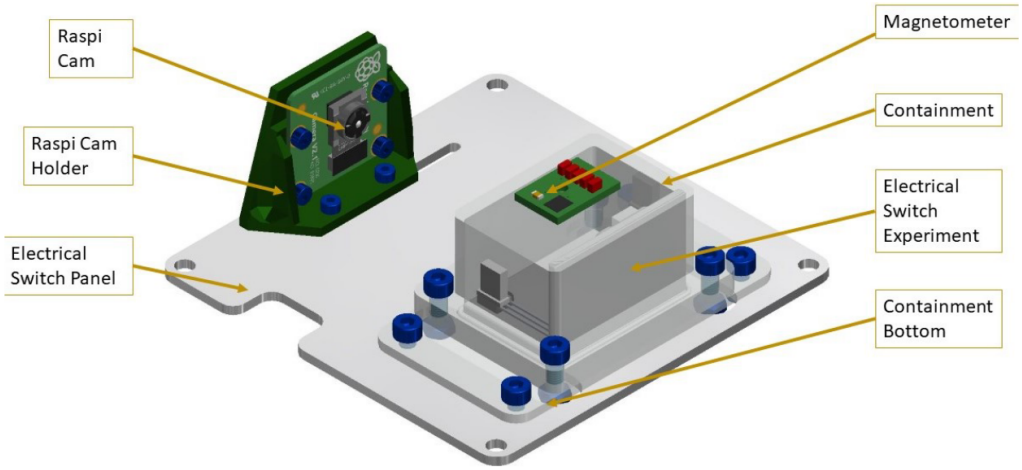


**Figure 5.2:** Rendered image of the electrical switch experiment [14]

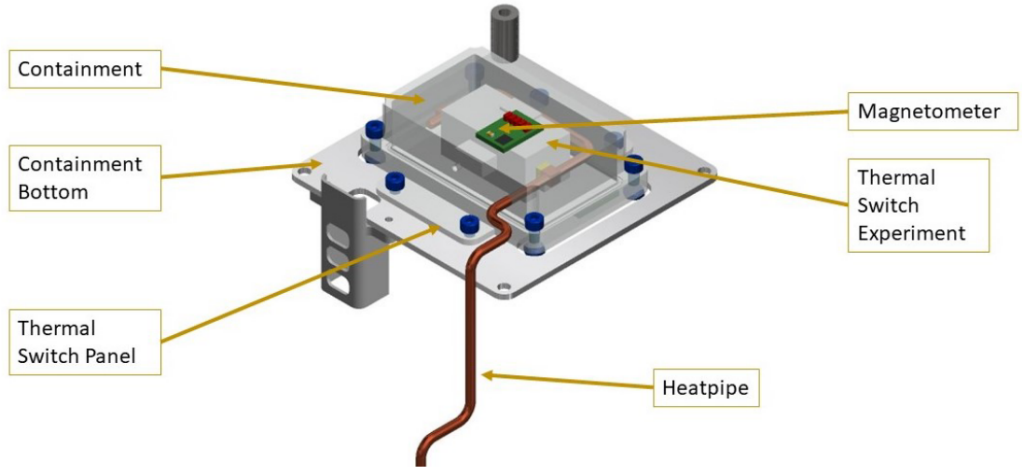


**Figure 5.3:** Rendered image of the thermal switch experiment [14]

## 5.3 Control Structure

With the information about the components of FARGO and their tasks, in addition to the previously defined system boundary, the control structure can be constructed. It follows a top to bottom hierarchy of control starting with the science sub-team of the FARGO project on top and ending with the physical experiment parts on the bottom. As usual in STPA this hierarchy implies no obedience.

This aspect of sending out a command, or feedback respectively, but not being sure if it gets received and carried out in the expected way is visualized in the control structure of FARGO, which can be seen in Figure 5.4, by placing the inscription of the arrows in it close to their origin. It can be seen that the commands from the science team for FARGO need to go through a communication interface. This interface is provided by Space Tango and should work in the command direction just as a relay. In the feedback direction it is more than just a communication relay as there is external telemetry data like power usage, which is collected by Space Tango, added to the results and internal telemetry data coming from FARGO. The command and feedback part between the communication interface and FARGOs main OBC is not controllable by the team. Like the interface it is provided and under control of Space Tango. The only control the FARGO team has is to implement the interfaces according to the specification. Everything inside the FARGO box is designed, tested and integrated by the team and thus allows for complete control. In addition to relaying the commands for the experiment runs, like which speeds to set to the ACS and which states to set to the switches, the main OBC monitors the power draw of the subsystems and gathers internal telemetry data like the temperature inside the module. It is able to cut the power to either subsystem in case the power draw is too high or the subsystem is unresponsive and needs to be reset. The feedback from the subsystem controllers is aggregated with the internal telemetry and provided to the communication interface, which polls the data. Further down the command hierarchy there are the subsystem controllers, which actuate the experiment hardware and collect the actual experiment data. The BLDC controller and the SBC for the ACS subsystem, both mentioned in the previous chapter, are shown as a single “ACS Controller”.

## 5.4 Identify UCAs

Having the control structure enables identifying UCAs, they are listed in Table 5.1. It can be seen that each control action has multiple options to be unsafe, depending on if it is given or its timing. For the set speed control action four possibilities to be unsafe have been identified. Not providing it (UCA-1) hinders collection and therefore provisioning of data as the ACS experiment cannot run. However providing it can be unsafe as well in case the speed setting is inappropriate (UCA-2). Inappropriate in this case could be a too high speed in the wrong direction, resulting in too much power draw, as rotating at higher speeds or “braking” by trying to rotate in the other direction too fast draws more power than a slow increase in rotation speed. Even with the correct direction and sufficiently small increments, so that the power draw is not too large, the rotational speed could get too high and thus the ACS would store rotational energy above the safety limit. The time to provide the set speed control action needs also to be correct. While providing it too late is not really unsafe, as it just results in unnecessary and unusable sensor data, providing it too early imposes the risk of the sensors not being ready to collect data (UCA-3). Providing it out of order with the ACS power on/off action might result in trying to control an inactive system (UCA-4), leading to the same results as not providing the control action at all. The set states control action given by the main OBC to the switches subsystem is similar to the set speed control action of the ACS subsystem, except for the providing part, which is in this case not unsafe. Due to being another control action and another subsystem the items are kept separate with their own set of IDs (UCA-5, UCA-6, UCA-7). For the set state on/off control action, given by the switch controller to the respective physical switches, the not providing (UCA-8) and providing too early (UCA-9) items are similar to the items in the set

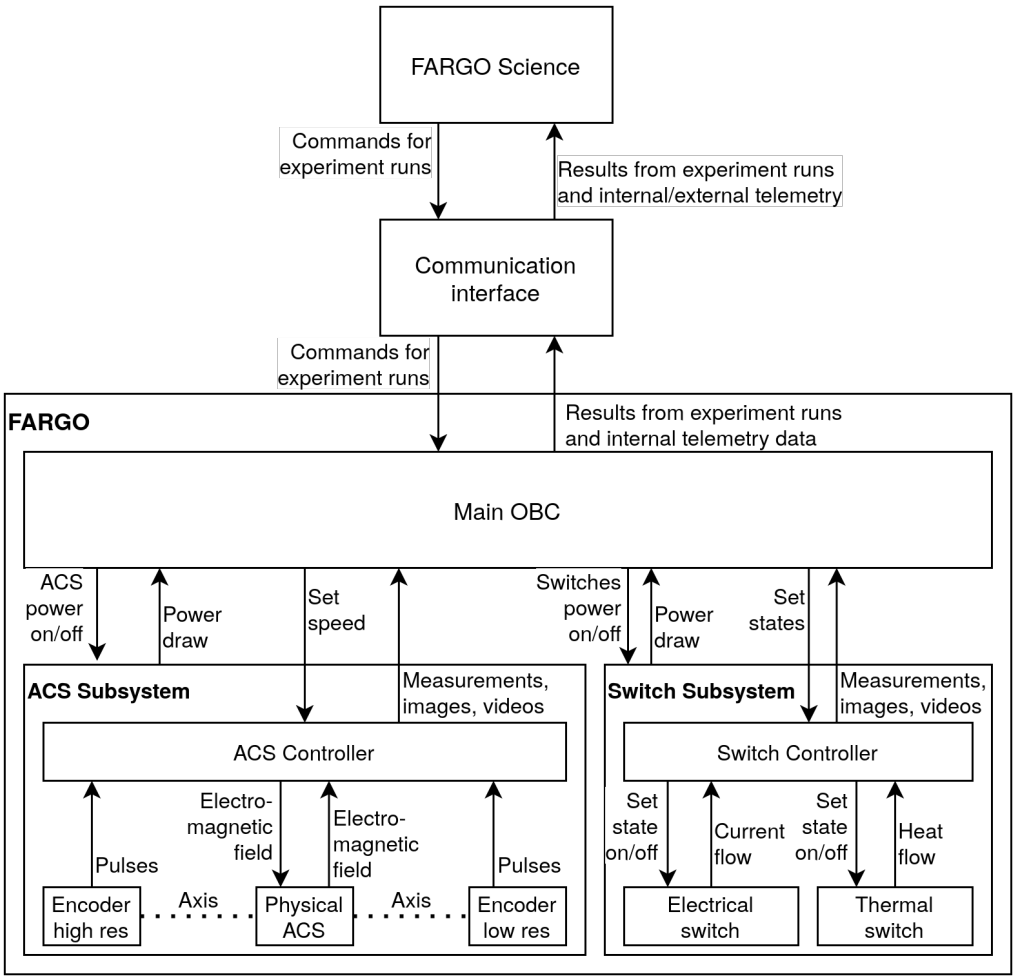


**Figure 5.4:** Control structure of FARGO

speed or set states control actions. Unique to this control action is that it can be applied too long (UCA-10), which could lead to damage on the EPMs, therefore rendering the switches inoperable and thus destroying this part of the FARGO experiment.

## 5.5 Identify Causal Factors

With the UCAs identified the causal factors or scenarios that might lead to providing an UCA can be investigated. Obviously hardware failures are a causal factor for not providing a control action, as a broken or damaged controller is unable to issue control actions correctly. This includes interrupted cable connections and other damaged electronic parts. The not providing part from the switch controller or out of order issues (from all control actions) can also appear if the respective subsystem is not powered. Having the experiment or parts of it unpowered can be the result of damaged electronics or a blown fuse or an error in the software. Naturally an issue with the software can lead to all of the mentioned unsafe actions. Even with the software working correctly as intended, which is in this case giving the user the most unrestricted control over how the sensor, actuators

**Table 5.1:** Unsafe Control Actions of the FARGO experiment

| **Control Action** | **Not providing** | **Providing** | **Too early, too late, out of order** | **Stopped too soon, applied too long** |
|---|---|---|---|---|
| Set speed | UCA-1: Main OBC does not provide the set speed control action [H-2, H-3] | UCA-2: Main OBC provides set speed control action with inappropriate speed [H-4, H-5] | UCA-3: Main OBC provides set speed control action before sensors are ready [H-2, H-3]<br><br>UCA-4: Main OBC provides set speed control action before ACS power on or after ACS power off control action [H-2, H-3] | - |
| Set states | UCA-5: Main OBC does not provide the set states control action [H-2, H-3] | - | UCA-6: Main OBC provides set states control action before sensors are ready [H-2, H-3]<br><br>UCA-7: Main OBC provides set states control action before switches power on or after switches power off control action [H-2, H-3] | - |
| Set state on/off | UCA-8: Switch controller does not provide the set state on/off control action [H-2, H-3] | - | UCA-9: Switch controller provides set state on/off control action before sensors are ready [H-2, H-3] | UCA-10: Switch controller applies the set state on/off action too long [H-2, H-3, H-5] |

and experiment computers should act, the unsafe control actions are not avoided, but the cause shifted from a programming mistake to a usage mistake. This is to be considered especially in terms of the set speed control action for the ACS, but also for the other control actions. Another factor that could upset the electronics, and thus endanger the mission, is electromagnetic interference (EMI) from the various magnets involved in the experiment. Although FARGOs objective is to test mechanical-free systems, there are prospects for mechanical failures that have an impact on mission loss. For example the ACS could get stuck during transport due to launch loads, which are hard to simulate, and therefore being unable to rotate.

## 5.6 Summary

In the last sections a STPA has been conducted on FARGO and identified several potential issues as results. These results are mainly compared to the software items in the verification matrix for FARGO, which are shown in Table 5.2. The complete verification matrix, in the version from the critical design review (CDR), can be found in Appendix B. In the original tables are four columns: an unique ID, a description of what needs to be verified, the methods used to verify the item and if it has been verified. For Table 5.2 the verified column has been omitted as it is used for progress tracking purposes and thus not relevant for the comparison. The ID of the items consists of three parts separated by dots. The first part indicates the subteam (also called subsystem) which is accountable for the item, where the letter is the first one of the subteams name. The used letters are M for mechanics, E for electronics and S for software. The second letter indicates from which kind of requirement the verification item is derived. Valid values for the kind of requirement source are F for functional, P for performance, D for design and O for operational requirements. The last part of the ID is a number incremented for each new item in the given subteam with a given requirement source to make the ID unique. Verification methods for the software items are test (T) and review of design (R). The comparison shows that the STPA found some possible issues that are already

**Table 5.2:** Software items from the verification matrix in the SED of FARGO

| ID | Description | Verification |
|---|---|---|
| S.P.1 | Software shall run stable at any time on the chosen electronic components. | R, T |
| S.D.1 | Software shall not include any single point of failure. | R, T |
| S.D.2 | The software shall support the used frequencies for experiment components, housekeeping, communication architecture and OBC. | R, T |
| S.O.1 | The system shall be able to perform a safety reboot of all experiments and the system itself at any time without endangering experiment behaviour. | R, T |
| S.O.2 | Software should ensure that a copy of the acquired data is stored within the experiment compartment. | R, T |

covered by the verification test, but it also found additional cases to be considered. Not getting any measurements would have been detected by the verification tests already in place, as their goal is making sure that the experiment is working properly. Power drawn from the supply lines and not exceeding their limits is part of the tests as well. It is covered by E.D.01 “Maximum currents

on each of the inface connectors three power lines must not be exceeded". Applying power too long to an EPM (UCA-10) can also be seen as already covered by the verification matrix. The relevant items are E.D.11 and E.D.13 stating that the electrical components shall not exceed their specifications and operate reliably over the whole mission time. The coil for switching an EPM is an electrical component and also the one which would be damaged by applying power for too long. Not covered by the verification test are experiments starting to run without the sensors being ready. Depending on the time between experiment start and sensors ready and recording this could be noticed as not getting measurements or not be noticed at all. So there is a chance that this issue is detected by the test, but also one that the test are passed with the issue being present. Having the control actions out of order is similar to this. An issue without even the chance of being detected by the verification tests is the ACS violating the RPM limit. In this constellation the STPA was able to detect issues that would not have been detected without it.

# 6 BDD Scenarios and Tests

In this chapter the results of the STPA performed in the previous chapter are used to derive BDD scenarios and tests for FARGO. Issues that are not solvable or not related to software, like hardware failures, are thereby ignored. First the UCAs are used to define BDD scenarios on the system level. After that options on how to apply these scenarios to FARGO are investigated. At the end the research question "which parts of the code are covered by the derived scenarios ?" is answered in the summary sub-section.

## 6.1 Deriving BDD Scenarios

To get from an UCA to BDD scenarios the UCA, in combination with the hazards it can cause and the causal factors that can lead to the UCA, is used as the narrative. From the narrative the actual scenarios are developed. To achieve this the hazards and causal factors are used for the context part of a scenario, which is labeled with **Given**. The UCA itself is the trigger or event part for the behavior and therefore gets the label **When**. How the system should behave is labeled with **Then** and needs to be a safe control action. The **When** and **Then** parts of the behavior can be extended for more context or more control actions to happen with the **and** keyword, which can be used to get more precise scenarios and thus more fine granular testing. Another option to extend the scenarios and tests is to not limit the **When** part to unsafe control actions, like it is the case for classic BDD without the STPA part. This extension option can be particular useful in case a control actions safety depends on the parameter it is executed with.

### 6.1.1 Set Speed Scenarios

For the set speed control action four UCAs have been found (see Table 5.1) which are now used to define BDD scenarios that in turn are used to test the software of FARGO.

1. **Not Providing**

   The scenario for not providing the set speed control action (UCA-1) looks like this:

   Narrative:
   UCA-1: Main OBC does not provide the set speed control action [H-2, H-3]

   *Scenario: experiment run is scheduled for the ACS to get data*
   **Given** an experiment run of the ACS is conducted
   **When** the main OBC does not provide the set speed control action
   **Then** an error is logged
   **and** the experiment run is marked as not completed

The reason to test for not providing and logging an error along with marking the experiment run as not completed is that distinguishing between *experiment ran as expected but there was no data to record* and *the experiment did not run as expected and therefore no data was recorded* needs to be enabled. Not having this differentiation would lead to less detailed results of the experiment.

2. **Providing**

   For "the providing" the set speed control action (UCA-2) the scenario is a little more involved:

   Narrative:
   UCA-2: Main OBC provides set speed control action with inappropriate speed [H-4, H-5]

   *Scenario: speed is too high*
   **Given** the ACS subsystem is powered
   **and** the sensors are ready
   **When** the set speed control action is provided with speed > speed limit
   **Then** the ACS does not change speed
   **and** an error is logged
   **and** the experiment run is marked as completed with errors

   *Scenario: speed is not too high*
   **Given** the ACS subsystem is powered
   **and** the sensors are ready
   **When** the set speed control action is provided with speed <= speed limit
   **Then** the ACS runs at the given speed

   The scenario where the speed to set is not too high is not required when the deriving process of STPA-BDD is followed strictly. Including it offers additional differentiation options though, especially in combination with the scenario for not providing the option. For example stopping the ACS by setting the speed to zero and not having this additional scenario could only be detected by decreasing (or increasing, depending on the direction) speed, which is fine unless the command is issued to define a baseline for an experiment run before sensor recordings are started.

3. **Too Early**

   Providing the set speed control action too early (UCA-3) gives this scenario:

   Narrative:
   UCA-3: Main OBC provides set speed control action before sensors are ready [H-2, H-3]

   *Scenario: experiment run is started*
   **Given** the ACS subsystem is powered
   **and** the sensors are not ready
   **When** the set speed control action is provided with speed <= speed limit
   **Then** the execution of it is delayed until the sensors are ready
   **and** a warning is logged
   **and** the experiment run is marked as completed with warnings

Getting a warning in case the intent is clearly stated and the only reason for the experiment run not being nominal is that the timing between sensor recording start and experiment start is not perfect is helpful in several ways. Perfect timing of sensor recording start and experiment start is not enforced but at the same time it is not required to start early with the recording and therefore record unnecessary data to avoid missing potentially important data. Over several experiment runs the timing also can be improved until it is optimal. This improvement can happen without the risk of lost data. Recording unnecessary data can be avoided by starting too early on purpose and move to the optimum from there.

4. **Out of Order**

   The scenarios for providing the set speed control action out of order with the ACS power on/off control action (UCA-4) are:

   Narrative:
   UCA-4: Main OBC provides set speed control action before ACS power on or after ACS power off control action

   *Scenario: permutation in the commanding*
   **Given** the ACS subsystem is unpowered
   **and** the ACS is expected to be unpowered
   **When** the set speed control action is provided with speed <= speed limit
   **Then** the ACS is powered on
   **and** the ACS runs at the given speed
   **and** a warning is logged
   **and** the experiment run is marked as completed with warnings

   *Scenario: the ACS is unpowered but should be powered*
   **Given** the ACS subsystem is unpowered
   **and** the ACS is not expected to be unpowered
   **When** the set speed control action is provided with speed <= speed limit
   **Then** an error is logged **and** the experiment run is marked as not completed

   An error in the commanding is covered by the first scenario and should not delay the experiment run and therefore save overall experiment time. Not having this could result in less experiment runs being made as the mission time is limited. For the second scenario the assumption is that the ACS is powered on from a previous experiment run and the ACS power on control action was not given on purpose therefore. This can be a favorable approach as the SBC controlling the ACS takes some time to boot and there is no necessity to reboot it between two experiment runs. However it is possible that an experiment run trips the fuse for the ACS subsystem unexpectedly. Handling this apparently command action out of order scenario like there was a permutation in commands or always issuing power on command actions before set speed control actions, even when they are not needed as the subsystem should be powered on, thus making power on a part of set speed, disclaims an easier insight into ACS behavior. Of course the tripped fuse would not go unnoticed as it will be logged, the relation to the experiment run would be less apparent however.

## 6.2 Multiple Controllers

Before the BDD scenarios can be used to validate the software for FARGO some more considerations need to be made on involved controllers. As already hinted during the STPA FARGO has multiple computers and $\mu$Cs (see Chapter 5.2). This makes the application of BDD scenarios to the software non-trivial as the straightforward way of translating them into test cases is not possible. In setting the ACS speed two $\mu$Cs, one SBC and three physical entities are involved in the FARGO cube lab. Additionally the commanding to set the ACS speed needs to be created somehow and sent to FARGO. Focusing on the cube lab there is the Main OBC which gets the commanding for an experiment run. In this commanding is the set speed control action, which gets relayed to the SBC part of the ACS controller after the ACS subsystem is powered on. The SBC is responsible for collecting the sensor data, therefore it either needs to report the sensors as ready to the Main OBC or the set speed command needs to be cached until the sensors are ready. Referring to the scenario for *too early* rules out the command caching option as it would create warnings all the time, which would render them useless. On the other hand allowing the command to be cached, therefore shifting the responsibility for the exact start time to the SBC, would eliminate UCA-3, but require a change in the BDD scenario. With the sensors being ready the SBC sends the set speed command to the $\mu$C of the BLDC controller which then creates the electromagnetic field that turns the physical ACS. The two encoders are turned by the physical ACS and report pulses back to the SBC, which can calculate the rotation speed from them. A third speed measurement is provided to the SBC from the BLDC controller. This measurement is calculated by the BLDC controller using the electromagnetic field the turning physical ACS imposes on it.

The actual speed at which the ACS should run has been omitted in this illustration of the inner workings of FARGO. It can be seen that there are multiple options where a speed check could take place. Referring to the BDD scenario either the Main OBC or the ACS SBC needs to check if the given speed violates the limit and then refuse to do the experiment run. Another option is to check, and modify if needed, the commanding before it is sent to FARGO thus eliminating UCA-2. This elimination can also be achieved by implementing the speed limit as the maximum speed the BLDC controller can provide. Accounting for the possibility that the speed limit might change during the mission in either direction putting a hard, not changeable limit on the ACS speed is suboptimal. Relying on the enforced, but now too high, speed limit lets a gone UCA reappear, while a too low limit constrains the experiment unnecessarily. Having this limit in a commanding check allows for simple adaption should the limit change.

To make the BDD scenarios applicable they either need to be split up and refined so that there are BDD scenarios for each involved controller, which can be straightforward translated to test cases, or they are translated into a black-box test, in which the whole FARGO cube lab is seen as black box.

## 6.3 Summary

It can be seen that deriving BDD scenarios from STPA results is straight forward and that it is trivial, but very beneficial, to include proper safe control actions in the deriving process. The application of these scenarios to the given system can be difficult though. To get evidence on which parts of the code are covered by the derived scenarios it is inevitable to refine them so that each involved controller can be tested individually. Using them unrefined and conducting black-box tests allows

for system testing but it cannot be known what the individual controllers do, therefore no conclusion about code coverage is possible in this case. Coverage can be increased by the inclusion of proper safe control actions, which allows for more thorough testing compared to only testing the code paths involved in (possible) unsafe control actions.

# 7 Software Validation

After defining the relevant BDD scenarios and noticing that there are several options to derive actual test cases from them in the last chapter an approach on how to apply this on FARGO is developed in this chapter.

## 7.1 Commanding

Starting at the top of the control hierarchy it can be seen that the FARGO science team needs an option to send commands for experiment runs. This commanding was mentioned in the previous chapter as a point to apply constraints to. Referring to the ICD, which can be found in [42], general requirements for the commanding are defined by Space Tango. In this general requirements the package size for a single package is 400 bytes. If the commanding fits not in such a single package it is split up in multiple 500 byte packages. Both, a commanding being sent as well as the detection of multiple packages being sent, need to be accounted for by Main OBC. How such a commanding looks like falls in the freedom of design that is given to the experiment developers. An option to reduce possibilities for errors is to not use the package split path, therefore all possible commanding options need to fit in one package each thus not being larger than 400 bytes. As crafting such a commanding byte code byte by byte is cumbersome, some sort of compiler is required to translate a simple, easy to understand high level commanding language into the bytes to be sent. The speed check to handle the possible unsafe set speed control action can be included in this compiler. A test would try to compile a commanding with a speed above the limit, which should result in an error and no commanding byte code being created. Checking for command not provided or provided out of order could be added as well, but should only issue a warning and still produce byte code. This would be beneficial to notice errors in the commanding early. The reason for only warn but still compile is that both, (apparent) out of order and not providing, need to be given on purpose. Not providing can be used to perform a status check by just gathering sensor data without an experiment run and (apparent) out of order might be needed for some experiment runs. The too early variant cannot be detected in this stage as it is dependent on the inner workings of FARGO, such as self-checks that might need different amounts of time or trying to reset an unresponsive sensor and therefore needing more time until the sensors ready stage.

## 7.2 Component Tests

For the component tests the commanding compiler is assumed to work correctly and the set speed control action with a speed above the limit is assumed as non-existent therefore. The complete experiment run for the given subsystem is assumed to be relayed to the respective subsystem SBC

to have a less tight coupling between the controllers. This seemingly eliminates UCA-3 but requires adaption of the BDD scenario. Actually UCA-3 is shifted from sitting between the Main OBC and the ACS controller to being inside the ACS controller, more specifically in the SBC part of it.

### 7.2.1 Communication Interface

As the communication interface is provided by Space Tango there is less testing needed but more training required on how to use it. According to the ICD it is a web portal showing data and enabling some interaction like initiating a commanding. For training and experiment testing purposes the team has been provided with an emulator. The experiment is connected to this emulator and then can be used as it were already on the ISS. To use it at least a partly working Main OBC is required. Parts that need to work are sending out results, which can consist of fake data, and a way of showing what commanding byte code it has received.

### 7.2.2 Main OBC

Given a proper commanding, which is assumed for testing the Main OBC, the BDD scenarios for "out of order" and "not providing" can be tested. The Main OBC can either be connected to the emulator or get the commanding directly from a test PC. The outputs of the Main OBC are then measured with an oscilloscope and logic analyzer to check if the order of provision and commands send down the control hierarchy are as expected by the scenarios.

### 7.2.3 ACS SBC

Testing the SBC of the ACS subsystem is more involved compared to testing the Main OBC. The part for the commanding input and commanding output to the BLDC controller is similar. To gather data about the experiment run there are several sensor inputs to the SBC as well. For a first pure behavioral test of the software these sensor inputs are faked with data in the expected range as well as outside of it. Then the real sensors are connected to the inputs and it is checked if the behavioral tests still succeed.

### 7.2.4 BLDC Controller, Physical ACS and Encoders

The BLDC controller, physical ACS and encoders are tested together, mostly due to the latter two being one physical entity when assembled. As the high resolution encoder has pretty tight tolerances and testing the encoders requires constant and repeatable rotation, the simplest solution is to turn the ACS via the BLDC controller. This also has the benefit of the BLDC controller being tested.

### 7.2.5 UCA-10

While the unsafe control actions have been put into BDD scenarios or are similar to ones that were, UCA-10 has been omitted and is also not similar to handled UCAs. During developing the EPMs and the circuit to actuate them it turned out that the power supply for the whole FARGO experiment is unable to provide sufiecient amperage to switch an EPM which has sufficient magnetic strength to be used for the switch experiment. To overcome this some buffering was built into the actuation circuit. This buffering can not hold more energy than needed to actuate the EPM, so applying too long is no longer unsafe. Safety was not the only reason for limiting the buffer capacity, a buffer storing more energy would not fit mechanically in the space available in the electronics bay.

## 7.3 End-to-End Test

It can be seen that the components can be tested individually, which is one possible way of applying the derived scenarios as stated in the previous chapter. The other stated way is to give commands to the complete FARGO experiment and check its outputs for correct behavior according to the derived scenarios. This is conducted with FARGO after all the components have been successfully tested individually and been put together. An additional improvement can be achieved by combining this end-to-end test with the individual tests in such a way that each successfully tested component is added to the overall system step by step, so component integration issues are detected and can be addressed immediately. Once FARGO has passed the end-to-end test and the flight readiness review (FRR), as part of regular space component qualification, it is ready for operation on the ISS.

## 7.4 Summary

Both ends of the spectrum of how the derived BDD scenarios can be applied have shown to be usable and a combination is even more optimal. In addition the individual component prior to the complete system test is a perfect fit for the V-model working process of FARGO. Further can be seen that during the implementation some design optimizations can occur and usually will be implemented. In the agile environment, for which STPA-BDD is proposed, this is not an issue as there are recurring STPA and BDD derivation steps over the sprints. For non-agile working processes either late design changes must be forbidden or some form of re-evaluation needs to be integrated. In the space world the concept of the “design freeze” exists.

# 8 Conclusion

This work started with the assumption that STPA-BDD is a ready for use tool to define software tests for safety-critical systems. To confirm this STPA-BDD was used to develop the software tests for the ISS-experiment FARGO. As expected for projects that undergo STPA for the first time, issues have been detected, which would have been overlooked otherwise. Then the BDD scenarios were derived from the results of the STPA. The derived scenarios could not be translated into test cases directly as multiple controllers are involved in the UCAs. To overcome this the scenarios can be refined so that each controller has its own scenarios or the scenario could be translated into a black-box test. Using both is a perfect fit for the validation in the V-model process that is used in FARGO, for which an approach is developed subsequently. During implementing this it was noticed that design improvements, which occurred hereby, need to be discarded if the BDD scenarios are set in stone. As this might work on paper, in reality it is more likely that the process would be ignored at this step and the improvements be built in anyway. To solve this the BDD scenarios are not fixed in our case, so that the process allows for including the improvements. Of course the scenarios are adapted to the improvements so that BDD validation remains possible. With this insights the initial hypothesis of STPA-BDD being a ready to use tool is improved. The refined assumption is that technically STPA-BDD is ready to use, but it can be enhanced to be usable with less obstacles.

# 9 Future Work

With this research it was shown that STPA-BDD, a concept developed for S-scrum, an agile process, technically can be used in non-agile processes, specifically in the V-model. It is only a proof-of-concept at this point in time though. To gain more insight sort of a replication of this work would be helpful. Modifications for the replication could include to use STPA-BDD over the whole project time instead of using it in the middle of a project as it was performed here or using a less involved project. The latter would allow for optimizing the BDD scenario derivation and their translation into actual test cases. From there this could be extended to more complex projects with the end goal of having a STPA-BDD handbook.

# Bibliography


[1] *Home Page - Space Tango*. URL: https://spacetango.com/ (visited on 09/30/2022) (cit. on p. 11).

[2] M. Irshad, R. Britto, K. Petersen. "Adapting Behavior Driven Development (BDD) for Large-Scale Software Systems". In: *Journal of Systems and Software* 177 (2021), p. 110944. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2021.110944. URL: https://doi.org/https://doi.org/10.1016/j.jss.2021.110944 (cit. on p. 11).

[3] M. Irshad, J. Börstler, K. Petersen. "Supporting Refactoring of BDD Specifications-An Empirical Study". In: *Information and Software Technology* 141 (2022), p. 106717. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2021.106717. URL: https://doi.org/https://doi.org/10.1016/j.infsof.2021.106717 (cit. on p. 11).

[4] B. D. Owens, M. S. Herring, N. Dulac, N. G. Leveson, M. D. Ingham, K. A. Weiss. "Application of a Safety-Driven Design Methodology to an Outer Planet Exploration Mission". In: *2008 IEEE Aerospace Conference*. Mar. 2008. DOI: 10.1109/aero.2008.4526677. URL: https://doi.org/10.1109/aero.2008.4526677 (cit. on p. 11).

[5] S. Wagner, B. Schatz, S. Puchner, P. Kock. "A Case Study on Safety Cases in the Automotive Domain: Modules, Patterns, and Models". In: *2010 IEEE 21st International Symposium on Software Reliability Engineering*. Nov. 2010. DOI: 10.1109/issre.2010.31. URL: https://doi.org/10.1109/issre.2010.31 (cit. on p. 11).

[6] S. Chokkadi, Y. Jeppu. "Teaching Stpa and Opm To Engineering Students - Industry Academia Experiences". In: *INCOSE International Symposium* 29.S1 (2019), pp. 17–27. DOI: 10.1002/j.2334-5837.2019.00666.x. URL: https://doi.org/10.1002/j.2334-5837.2019.00666.x (cit. on p. 11).

[7] R. Patriarca, M. Chatzimichailidou, N. Karanikas, G. D. Gravio. "The Past and Present of System-Theoretic Accident Model and Processes (STAMP) and Its Associated Techniques: a Scoping Review". In: *Safety Science* 146 (2022), p. 105566. DOI: 10.1016/j.ssci.2021.105566. URL: https://doi.org/10.1016/j.ssci.2021.105566 (cit. on p. 11).

[8] A. Dong et al. "Application of CAST and STPA to railroad safety in China". PhD thesis. Massachusetts Institute of Technology, 2012 (cit. on p. 11).

[9] Y. Song. "Applying system-theoretic accident model and processes (STAMP) to hazard analysis". PhD thesis. 2012 (cit. on p. 11).

[10] Y. Wang, S. Wagner. "Combining STPA and BDD for Safety Analysis and Verification in Agile Development: A Controlled Experiment". In: *Lecture Notes in Business Information Processing*. Lecture Notes in Business Information Processing. Springer International Publishing, 2018, pp. 37–53. DOI: 10.1007/978-3-319-91602-6_3. URL: https://doi.org/10.1007/978-3-319-91602-6_3 (cit. on pp. 11, 19).

[11] Y. Wang, D. R. Degutis, S. Wagner. “Speed up BDD for safety verification in agile development”. In: *Proceedings of the 19th International Conference on Agile Software Development: Companion*. May 2018. DOI: 10.1145/3234152.3234181. URL: https://doi.org/10.1145/3234152.3234181 (cit. on pp. 11, 18).

[12] Y. Wang. “System-Theoretic Safety Analysis in Agile Software Development”. In: (2018). DOI: 10.18419/opus-10118. URL: https://doi.org/10.18419/opus-10118 (cit. on pp. 11, 17, 18).

[13] *FARGO - KSat e.V.* URL: https://www.ksat-stuttgart.de/en/fargo-en/ (visited on 05/02/2022) (cit. on pp. 11, 14).

[14] S. S., B. D., E. M., S. F., O. M., H. N., K. C., D. J., T. F., G. S., A. D., H. S., S. M., R. Y., K. P., K. M., W. B., Z. S., P. D., B. M., K. B., S. M., G. E., B. L., H. G., F. S. “FARGO - Validation of Space-relevant Ferrofluid Applications on the ISS”. In: *71st DLRK*. Dresden, Germany, 2022 (cit. on pp. 11, 16, 24, 25).

[15] F. Schäfer, M. Ehresmann, S. Zajonz, S. Grossmann, E. Guiterrez, N. Heinz, M. O’donohue, C. Korn, G. Herdrich, J. Gente, D. Gkoutzos, D. Levi, S. Weikert, A. Wiegand. “Ferrofluid-based attitude control for small satellites”. In: Sept. 2022 (cit. on p. 11).

[16] M. Ehresmann, D. Bölke, S. Hofmann, F. Hild, K. Grunwald, S. Sütterlin, C. Behrmann, N. Heinz, G. Herdrich, R. Jemmali. “Experiment Results and post-flight Analysis of the ISS Student Experiment PAPELL”. In: Oct. 2019 (cit. on pp. 11, 14).

[17] T. Ishimatsu, N. G. Leveson, J. Thomas, M. Katahira, Y. Miyamoto, H. Nakao. “Modeling and Hazard Analysis Using Stpa”. In: *Proceedings of the 4th IAASS Conference, Making Safety Matter* (2010). Online; accessed 27 August 2022. URL: http://hdl.handle.net/1721.1/79639 (cit. on p. 13).

[18] N. G. Leveson. “STPA: A New Hazard Analysis Technique”. In: *Engineering a Safer World: Systems Thinking Applied to Safety*. Engineering a Safer World. The MIT Press, 2012. DOI: 10.7551/mitpress/8179.003.0013. URL: https://doi.org/10.7551/mitpress/8179.003.0013 (cit. on p. 13).

[19] J. Smart. *BDD in Action: Behavior-driven development for the whole software lifecycle*. Simon and Schuster, 2014 (cit. on p. 13).

[20] *ARISE - Home*. URL: https://www.uni-due.de/physik/arise/ (visited on 05/04/2022) (cit. on p. 14).

[21] *EXCISS – Experimental Chondrule Formation at the ISS*. URL: https://www.goethe-university-frankfurt.de/67786978/EXCISS___Experimental_Chondrule_Formation_at_the_ISS (visited on 05/04/2022) (cit. on p. 14).

[22] J. Weppler, C. Lemack, T. Steinpilz, G. Musiolik, M. Kruss, T. Demirci, F. Jungmann, A. Krämer, J. Tappe, M. Aderholz, J. Teiser, G. Wurm, T. Koch, Y. Schaper, R. Nowok, A. Beck, O. Christ, P.-T. Genzel, M. Lindner, G. Herdrich. “Überflieger - A Student Competition for ISS Experiments”. In: Sept. 2017 (cit. on pp. 14, 21).

[23] S. Sütterlin, N. Heinz, F. Hild, K. Grunwald, M. Hell, S. Hofmann, P. Ziegler, C. Korn, M. Schneider, F. Frank, M. Ehresmann, G. Herdrich, D. Helmer. “PAPELL: Experiments, Prototyping and Mechanical Engineering”. In: Apr. 2018 (cit. on p. 14).

[24] K. Grunwald, F. Hild, S. Sütterlin, N. Heinz, A. Aslan, F. Grabi, M. Sauer, R. Schweigert, M. Ehresmann, G. Herdrich. "PAPELL: Student team lead tenets and challenges in an international project". In: Apr. 2018 (cit. on p. 14).

[25] M. Ehresmann, K. Grunwald, S. Aslan, F. Grabi, R. Schweigert, P. Ziegler, M. Hell, M. Schneider, F. Frank, C. Korn, A. Causevic, K. Waizenegger, A. Behnke, V. Hertel, P. Sahli, D. Bölke, M. Siedorf, C. Behrmann, T. Ott, S. Hofmann. "PAPELL: SOLID-STATE PUMPING MECHANISM". In: Sept. 2018 (cit. on p. 14).

[26] *FerrAS - KSat e.V.* URL: https://www.ksat-stuttgart.de/en/ferras-en/ (visited on 05/02/2022) (cit. on p. 14).

[27] *PapellObcSoftware - Bitbucket*. URL: https://bitbucket.org/ksatstuttgart/papellobcsoftware/ (visited on 05/08/2022) (cit. on p. 14).

[28] A. Abdulkhaleq, S. Wagner, N. Leveson. "A Comprehensive Safety Engineering Approach for Software-Intensive Systems Based on Stpa". In: *Procedia Engineering* 128 (2015), pp. 2–11. DOI: 10.1016/j.proeng.2015.11.498. URL: https://doi.org/10.1016/j.proeng.2015.11.498 (cit. on p. 17).

[29] M. E. France. "Enineering for humans: a nex extension to STPA". In: Massachusetts Institute of Technology, 2017. URL: http://hdl.handle.net/1721.1/112357 (cit. on p. 17).

[30] M. S. Placke. "Application of STPA to the integration of multiple control systems : a case study and new approach". In: *nil*. Massachusetts Institute of Technology, 2014. URL: http://hdl.handle.net/1721.1/90170 (cit. on p. 17).

[31] S. Baumgart, J. Froberg, S. Punnekkat. "Can STPA be used for a System-of-Systems? Experiences from an Automated Quarry Site". In: *2018 IEEE International Systems Engineering Symposium (ISSE)*. Oct. 2018, nil. DOI: 10.1109/syseng.2018.8544433. URL: https://doi.org/10.1109/syseng.2018.8544433 (cit. on p. 17).

[32] H. Nakao, M. Katahira, Y. Miyamoto, N. Leveson. "Safety guided design of crew return vehicle in concept design phase using STAMP/STPA". In: *Proc. of the 5: th IAASS Conference*. Citeseer. 2011, pp. 497–501 (cit. on p. 17).

[33] J. M. Rising, N. G. Leveson. "Systems-Theoretic Process Analysis of Space Launch Vehicles". In: *Journal of Space Safety Engineering* 5.3-4 (2018), pp. 153–183. DOI: 10.1016/j.jsse.2018.06.004. URL: https://doi.org/10.1016/j.jsse.2018.06.004 (cit. on p. 17).

[34] F. Meyen, A. Krishnamurthy, J. Hoffman. "System Theoretic Process Analysis (STPA) of the Mars oxygen ISRU experiment (MOXIE)". In: *2018 IEEE Aerospace Conference*. Mar. 2018. DOI: 10.1109/aero.2018.8396586. URL: https://doi.org/10.1109/aero.2018.8396586 (cit. on p. 17).

[35] I. Alexander. "Misuse Cases: Use Cases With Hostile Intent". In: *IEEE Software* 20.1 (2003), pp. 58–66. DOI: 10.1109/MS.2003.1159030. URL: https://doi.org/10.1109/MS.2003.1159030 (cit. on p. 17).

[36] T. Y. Chen, S. C. Cheung, S. Yiu. "Metamorphic Testing: A New Approach for Generating Next Test Cases". In: *CoRR* abs/2002.12543 (2020). arXiv: 2002.12543. URL: https://arxiv.org/abs/2002.12543 (cit. on p. 17).

[37] J. Ryser, M. Glinz, et al. "Scent: a Method Employing Scenarios To Systematically Derive Test Cases for System Test". In: *Berichte des Instituts für Informatik* 3 (2000) (cit. on p. 17).

[38] M. Althoff, S. Lutz. “Automatic Generation of Safety-Critical Test Scenarios for Collision Avoidance of Road Vehicles”. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 1326–1333. DOI: 10.1109/IVS.2018.8500374. URL: https://doi.org/10.1109/IVS.2018.8500374 (cit. on p. 17).

[39] J. Hatakeyama, D. Seal, D. Farr, S. Haase. “Systems engineering "Vïn a model-based engineering environment: Is it still relevant?” In: *2018 AIAA SPACE and Astronautics Forum and Exposition*. 2018, p. 5326 (cit. on p. 19).

[40] R. K. Yin. *Case study research: Design and methods*. Vol. 5. sage, 2009 (cit. on p. 19).

[41] *ISS Safety Requirements Document*. URL: https://ntrs.nasa.gov/api/citations/20210009936/downloads/SSP%202051721-Baseline.pdf (visited on 09/30/2022) (cit. on p. 21).

[42] D. Z. für Luft- und Raumfahrt e. V. (DLR). *Überflieger2 8211; Studierendenexperimente auf der ISS*. URL: https://ueberflieger.space/ (visited on 09/29/2022) (cit. on p. 37).

# A Acronyms

**Table A.1:** Acronyms

| Acronym | Name |
|---|---|
| ACS | Attitude Control System |
| BDD | Behaviour-Driven Development |
| BLDC | Brush-Less Direct Current |
| CDR | Critical Design Review |
| DLR | German Aerospace Center |
| DPG | German Physical Society |
| EMI | ElectroMagnetic Interference |
| EPM | Electro Permanent Magnet |
| FARGO | Ferrofluid Application Research Goes Orbital |
| FRR | Flight Readiness Review |
| FerrAS | Ferrofluid Application Study |
| ICD | Interface Control Document |
| IMU | Inertial Measurement Unit |
| IRS | Institute for Space Systems at the University of Stuttgart |
| ISS | International Space Station |
| ISTE | Institute of Software Engineering at the University of Stuttgart |
| KSat e.V. | Small Satellite Student Society at the University of Stuttgart |
| OBC | On-Board Computer |
| PAPELL | Pump Application using Pulsed Electromagnets for Liquid reLocation |
| PCB | Printed Circuit Board |
| PR | Public Relations |
| PWM | Pulse Width Modulation |
| RF | Radio Frequency |
| RPM | Revolutions Per Minute |
| S-Scrum | Safe Scrum |
| SBC | Single Board Computer |
| SED | Student Experiment Document |
| STAMP | System-Theoretic Accident Model and Processes |
| STPA | Systems-Theoretic Process Analysis |
| UAT | User Acceptance Tests |
| UCA | Unsafe Control Action |
| $\mu$C | Micro Controller |
| $\mu$G | Micro gravity |

# B Verification matrix from FARGO SED

# 5 EXPERIMENT VERIFICATION AND TESTING

## 5.1 Verification Matrix

Four established verification methods (for details see: ECSS-E-ST-10-02C):

- Verification by test (T)
- Verification by inspection (I)
- Verification by analysis or similarity (A)
- Verification by review-of-design (R)

The following table lists all requirements and specifies their respective verification method(s).

**Table 5-1 Verification Matrix Functional Requirements**

| *ID* | *Description* | *Verification* | *Verified* |
|---|---|---|---|
| E.F.01 | The Electric Switch shall control a defined test current (on/off). | T | |
| E.F.02 | The Experiment SBC shall be able to turn the EPMs of both switch experiments on and off. | T | |
| E.F.03 | The voltage and current between the two contacts of the Electrical Switch shall be measured. | T, R | |
| E.F.04 | The temperature of the EPMs and the Electrical Switch shall be measured. | T, R | |
| E.F.05 | The magnetic field strength of the EPM of the Electrical Switch shall be measured. | T, R | |
| E.F.06 | The Thermal Switch should set a defined a heat flow between a heat source and a heatsink on and off. | T | |
| E.F.07 | The temperature shall be measured at several points of the heat conductor at the Thermal Switch assembly. | T | |
| E.F.08 | The magnetic field strength of the EPM in the Thermal Switch shall be measured. | T | |
| E.F.09 | The coils of the ACS-BLCD shall generate a rotating magnetic field to achieve acceleration and deceleration of the contents of the fluid containment chamber. | T | |
| E.F.10 | The ACS-BLDC SBC shall derive the rotation velocity of the rotor as well as | T, A | |

**Figure B.1:** Page 1 of the verification matrix from the FARGO SED

| ID | Description | Verification | Verified |
|---|---|---|---|
| | the rotational frequency of the magnetic field. | | |
| E.F.11 | Torque generated by the ACS shall be measured or derived from position data. | T, A, R | |
| E.F.12 | The temperature of the coils and the magnetic field of the ACS-BLDC shall be measured. | T, R | |
| E.F.13 | The Main Microcontroller shall be able to communicate through the CubeLab Interface Connector and with the Experiment SBC. | T | |
| E.F.14 | The Main Microcontroller shall be able to cut power to the experiments. | T | |
| E.F.15 | The Main Microcontroller shall be able to monitor the system's current draw and log the data. | T | |
| E.F.16 | The Main Microcontroller shall measure and shall log environmental conditions: temperature, acceleration, pressure, magnetic field. | T | |
| E.F.17 | Electronic fuses shall monitor the overall system's current draw and limit current to abide by restrictions stated in the ICD. | T | |
| E.F.18 | The ACS-BLDC SBC shall control one camera and save its video data. | T | |
| E.F.19 | The Switches SBC shall control one camera and save its video data. | T | |
| E.F.20 | The electrical switch experiment shall be recorded by a camera. | T, I | |
| E.F.21 | The ACS-BLDC experiment shall be recorded by a camera. | T, I | |
| M.F.01 | The Experiment structure shall not fail under the specified load cases. | A | A |
| M.F.02 | The ACS-BLDC shall apply torque by accelerating ferrofluid in a rationally symmetrical setup. | A, I | A, I |
| M.F.03 | The ACS-BLDC shall rotate freely. | T | |
| M.F.04 | The EPMs included in the Electrical Switch shall move the Galinstan between the electrical contacts in order to enable or prevent a current flow. | T | |

**Figure B.2:** Page 2 of the verification matrix from the FARGO SED

| ID | Description | Verification | Verified |
|---|---|---|---|
| M.F.05 | The EPMs included in the Thermal Switch shall move Galinstan between the contacts in order to enable and prevent a heat flow. | T | |

**Table 5-2 Verification Matrix Performance Requirements**

| ID | Description | Verification | Verified |
|---|---|---|---|
| *E.P.01* | The temperature of all experiments and housekeeping sensors shall be measured between 0 and 80 °C. | R | |
| *E.P.02* | The temperature of all experiments and housekeeping sensors shall be monitored within an accuracy of 0.5 °C. | T, R | |
| *E.P.03* | The temperature of all experiments and housekeeping sensors shall be monitored at a rate of at least 1 Hz. | T | |
| *E.P.04* | The magnetic field strength of all experiments shall be measured between 0 and 80 mT. | R | |
| *E.P.05* | The magnetic field strength of all experiments shall be monitored with an accuracy of 1 mT. | T, R | |
| *E.P.06* | The magnetic field strength of all experiments h shall be monitored at a rate of at least 2 Hz. | T | |
| *E.P.07* | The voltage of the Electrical Switch shall be measured between 0 and 12 V. | T, R | |
| *E.P.08* | The voltage of the Electrical Switch shall be monitored with an accuracy of 2 mV. | T, R | |
| *E.P.09* | The voltage of the Electrical Switch shall be monitored at a rate of 10 kHz. | T | |
| *E.P.10* | The torque of the ACS-BLDC system shall be measured between 0 and 100 mNm. | T | |

**Figure B.3:** Page 3 of the verification matrix from the FARGO SED

| ID | Description | Verification | Verified |
|---|---|---|---|
| *E.P.11* | The torque of the ACS-BLDC system shall be measured with an accuracy of 0.1 to 1 mNm. | A, R, T | |
| *E.P.12* | The angular velocity of the ACS-BLDC rotor shall be measured between 0 and 2000 RPM (1/min). | *T* | |
| *E.P.13* | The angular velocity of the ACS-BLDC actuator system shall be measured with an accuracy of 6 RPS (1/s). | R, T | |
| *E.P.14* | The angular position of the ACS-BLDC actuator shall be measurable for a full revolution of 360°. | T | |
| *E.P.15* | The angular measurements of the ACS-BLDC actuator shall be possible in both directions. Clockwise and counterclockwise. | R, T | |
| *E.P.16* | The angular position of the ACS-BLDC actuator shall be measured with an accuracy of at least100 arcsec. | R, T | |
| *E.P.18* | The camera of the Electrical Switch shall record at a resolution of 1280x720 pixels. | R | |
| *E.P.19* | The camera of the Electrical Switch shall record at 60Hz. | R | |
| *E.P.20* | The camera of the ACS-BLDC shall record at a resolution of 1280x720 pixels. | T, I | |
| *E.P.21* | The camera of the ACS-BLDC shall record at 60Hz. | T, I | |
| *S.P.1* | Software shall run stable at any time on the chosen electronical components. | R, T | No |

**Table 5-3 Verification Matrix Design Requirements**

| ID | Description | Verification | Verified |
|---|---|---|---|
| *E.D.01* | Maximum currents on each of the interface connector's three power lines must not be exceeded. | T | |
| *E.D.02* | Each RTN (Return Line) must be respectively connected to minimize ground loops. | R | |

**Figure B.4:** Page 4 of the verification matrix from the FARGO SED

| ID | Description | Verification | Verified |
|---|---|---|---|
| *E.D.03* | A class H Bond shall be measured between the CubeLab and Payload Card RTNs. | T | |
| *E.D.04* | A resistance of >1 MΩ shall be measured between the Power-RTN and the external Surface of the CubeLab module. | T | |
| *E.D.05* | All power wiring shall be PTFE (Teflon) coated and appropriately sized to 150% of the designated maximum current. | R | |
| *E.D.06* | Payload shall avoid using electrolytic capacitors.<br>Tantalum, aluminium and ceramic are acceptable. | R, I | |
| *E.D.07* | The operating voltage of the UART Tx/Rx pins shall not be exceeded. | T, R | |
| *E.D.08* | Monitoring the currents of all experiments shall be performed. | R | |
| *E.D.09* | The temperature of critical components shall be monitored with temperature sensors. | R, A | |
| *E.D.10* | All PCB shall be coated in silicone. | I | |
| *E.D.11* | All electrical components shall not exceed their specified parameters, even when used for long periods. | T, R | |
| *E.D.12* | The Electronics shall fit inside the intended compartment (boundary box). | I, R, T | |
| *E.D.12a* | All components shall withstand environmental conditions during launch, flight, operation and return. | R | |
| *E.D.12b* | All electrical components and the PCBs shall be mounted in a way to withstand launch loads and vibrations. | R | |
| *E.D.13* | All electrical components shall operate safely and reliably for the maximum duration outlined in the experiment timeline | R | |
| *M.D.1* | The three sub-experiments including their electronics, as well as all necessary structural parts must fit into the 2U CubeLab experiment container. | I | I |

**Figure B.5:** Page 5 of the verification matrix from the FARGO SED

| ID | Description | Verification | Verified |
|---|---|---|---|
| *M.D.2* | The experiment shall be easily accessible in all stages of integration until the TANGO CubeLab is closed. | R, I | R, I |
| *M.D.3* | The experiment structure should provide an acceptable maximum displacement of components in all load cases, which will be determined during the simulations. | A | A |
| *M.D.4* | The Precision frame for the ACS-BLDC should provide a defined precision of the two bearings to each other in all load cases defined in the TANGO ICD, 7.1, until re-entry which will be defined during ground tests. | A | A |
| *M.D.5* | The experiment structure shall always stay intact during all possible load cases defined in the TANGO ICD, 7.1, , with a minimum safety of 2. | A, R | A, R |
| *M.D.6* | All components must not have a mass of more than 4000 grams. | A, T | A |
| *M.D.7* | The Containment vessels of the ACS and the switches shall provide a tight seal during all load cases defined in the TANGO ICD, 7.1, and over all standing time until reopening. | T, R | R |
| *M.D.8* | The Containment vessels of the ACS and the switches shall be produced of a material providing a chemical resistance to all provided fluids for the full mission duration. | T, R | R |
| *M.D.9* | The Tango Module shall not heat up to a Surface Temperature greater as the one stated by TANGO. This will be checked. | A, R | R |
| *M.D.10* | The Precision frame shall hold the bearings of the ACS- BLDC in a tolerance to each other that will be determined during the test and that provides sufficiently low torque resistance. | T, R | R |
| *M.D.11* | The whole experiment must be assembled in a way that allows for a disassembly to a defined level at any given point in the integration, this level being down to the experiments. | T, R | T, R |

**Figure B.6:** Page 6 of the verification matrix from the FARGO SED

| ID | Description | Verification | Verified |
|---|---|---|---|
| *S.D.1* | Software shall not include any single point of failure. | T, R | No |
| *S.D.2* | The software shall support the used frequencies for experiment components, housekeeping, communication architecture and OBC. | T, R | In Progress |

**Table 5-4 Verification Matrix Operational Requirements**

| ID | Description | Verification | Verified |
|---|---|---|---|
| *E.O.1* | The experiment shall be totally without power within 20 s after being removed from the USB connection. | R, T | |
| S.O.1 | The system shall be able to perform a safety reboot of all experiments and the system itself at any time without endangering experiment behaviour. | R, T | No |
| S.O.2 | Software should ensure that a copy of the acquired data is stored within the experiment compartment. | R, T | In Progress |

## 5.2 Test Plan

********** Science **********

| | |
|---|---|
| ***Test number*** | *0.1* |
| ***Test type*** | *Functionality tests* |
| ***Test facility*** | *KSat workspace* |
| ***Tested item*** | *Electrical Switch* |
| ***Test level/procedure*** | *Functionality test to determine whether a setup of one EPM and one PM is able to move the Galinstan-ferrofluid reliably* |
| ***Test duration*** | *1 day* |
| ***Date / status*** | *August 2022 / completed* |
| ***Verified requirement*** | *E.F.01,* M.F.04 |

| | |
|---|---|
| ***Test number*** | *0.2* |
| ***Test type*** | *Functionality tests* |
| ***Test facility*** | *KSat workspace* |

**Figure B.7:** Page 7 of the verification matrix from the FARGO SED

**Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum, Unterschrift