A Mobile and Compact Control Center for Quick Decentral Satellite Access

Stefan A. Gärtner[1], Norbert Harder[2], Jens H. Hartung[3], Markus Hobsch[4], and Martin Weigel[5]
German Space Operations Center GSOC, DLR Oberpfaffenhofen, 82234 Weßling, Germany

Compact and inexpensive Earth observation satellites in low Earth orbit are now routinely developed by universities, "New Space" businesses, and space agencies. They enable new opportunities for fast turnaround times of imaging data takes, which is e. g. particularly important for disaster response. For this kind of satellites and the missions enabled by them a ground system exhibiting the same characteristics, namely being compact and mobile, yet inexpensive and flexible, is desired.

We present DLR's approach for the provisioning of a ground segment fit for the kind of missions outlined above. The objective of this project consists of the engineering, delivery, and demonstration of a compact and yet complete Mission Operations System, runnable on commodity mobile hardware, enabling fully automated workflow-driven operations of alike missions from anywhere in the world with access to a ground station or ground station network.

Just as disasters strike suddenly, the ground segment needs to be set up and spun up in a timely manner. This leads to the requirement of being able to quickly roll out the system on new hardware, possibly even several of these systems in parallel. Our paper provides insight on how we perform the automatic deployment and provisioning.

Because the system is supposed to be decentralized and used in the field, particular challenges need to be overcome resulting from the lack of all of the infrastructure typically present in conventional control centers, such as network connectivity. An embedded Flight Dynamics system is taking care of automated orbit determination and related event generation to support the mission needs and maneuver capabilities. Special effort is made to cope with auxiliary data that may not be updated on a regular basis in a closed mission environment.

The feasibility of the concept is demonstrated by a first system deployment as drop-in replacement for the existing conventional Mission Operations System for DLR's BIROS satellite at the GSOC control center. A second demonstration campaign is performed from a remote location without access to control center infrastructure.

Abbreviations

CCSDS	Consultative Committee for Space Data Systems
CI/CD	Continuous Integration/Continuous Deployment
COTS	Commercial Off-The-Shelf
DLR	German Aerospace Center/Deutsches Zentrum für Luft- und Raumfahrt
ERP	Earth Rotation Parameter
ESA	European Space Agency
FDS	Flight Dynamics System
GDS	Ground Data System
GECCOS	GSOC Enhanced Command and Control System for Operating Spacecrafts

[1] Mission Control and Data Systems Engineer, Mission Technology Department, DLR German Aerospace Center, Münchner Straße 20, 82234 Weßling, Germany, ORCID: `http://orcid.org/0000-0002-4077-9851`

[2] Mission Control and Data Systems Engineer, Mission Technology Department, DLR German Aerospace Center, Münchner Straße 20, 82234 Weßling, Germany

[3] Mission Planning Engineer, Mission Technology Department, DLR German Aerospace Center, Münchner Straße 20, 82234 Weßling, Germany

[4] Mission Control and Data Systems Group Head, Mission Technology Department, DLR German Aerospace Center, Münchner Straße 20, 82234 Weßling, Germany

[5] Flight Dynamics Engineer, Space Flight Technology Department, DLR German Aerospace Center, Münchner Straße 20, 82234 Weßling, Germany

GNSS	Global Navigation Satellite System
GSOC	German Space Operations Center
LEO	Low Earth Orbit
MCS	Monitoring and Control System
MIB	Mission Information Base
MPS	Mission Planning System
NCTRS	Network Control and TM/TC Router System
OD	Orbit Determination
ProToS	Procedure Tool Suite
SCOS	Satellite Control and Operation System
SLE	Space Link Extension
SSB	SLE Switch Board
TC	Telecommand
TLE	Two-Line Elements
TM	Telemetry
VM	Virtual Machine
XML	Extensible Markup Language

1. Introduction

One of the exciting new topics in space operations is the emerging market of small, inexpensive, flexible to use satellites which are deployed into low-Earth orbits. Ground segments for these missions should reflect this new paradigm by being equally compact, inexpensive and fast to roll out, easily maintainable, and providing straight-forward operability for small staff. Ideally, such a ground-segment can be operated right where the acquired payload data is needed most, for example in disaster areas of the world where quickly unfolding events demand timely access to information.

In this paper we show the design, implementation, and utilization of such a Mission Operations System and how it lends itself to a *compact* and *mobile* ground segment. The following high-level design drivers guide the development of the ground segment:

- The ground segment shall be movable between different places of operation.
- The ground segment shall operate independently, i. e. it shall not be deeply integrated into facilities and infrastructure and shall not rely on permanent Internet access.
- External interfaces (e. g. to ground stations) shall be standards-compliant if such standards exist.
- Copies and updates of the ground segment shall be able to be put into operation quickly.
- The ground segment shall be operated by one operator.
- The ground segment shall enable demonstration operations of the BIROS satellite's optical imaging payload.

1.1 Compactness and Mobility

Achieving *compactness* with respect to ground systems is made possible by the focus on specific mission types as outlined above: Satellites are single-purpose; therefore, no special operation modes need to be supported. Contingency handling is kept to a minimum, not only for ground but also for space segment contingencies. The remaining routine operations tasks are predisposed to automation. This in turn allows a workflow-driven operations approach that can be executed by a single operator on a single console. The drastically reduced number of consoles, the lack of redundancy and the single-purpose space segment allow the whole ground segment to be implemented on commodity hardware like a single laptop.

Compactness is necessary but not sufficient to achieve *mobility*. A ground segment is usually embedded into a larger infrastructure —the control center— that provides multi-mission services, ground station connectivity and facilities. The presented ground segment is built from components derived from the GSOC multi-mission tool suite. As such, they are ideally tailored to each other and can be integrated into a self-contained assembly with as little external dependencies as possible. Some multi-mission services need to be stripped down and integrated. Ground station connectivity still needs to be provided externally, albeit the system is setup assuming a standardized antenna interface like CCSDS SLE [1]. This allows plugging the ground segment in any ground station network or single antenna implementing the standard.

1.2 Demonstration Campaigns

It should be noted that the ground segment is not tailored to a specific mission but set up in a way to be adaptable to satellites belonging to the class of small, single-purpose LEO space vehicles as described above. Nonetheless, the whole concept is demonstrated in two campaigns currently in preparation with an existing satellite operated by GSOC, namely BIROS [2]. With its specifications[1] BIROS fits the class of satellites targeted by this ground segment. BIROS houses an optical imaging payload and we show the execution of an exemplary workflow by means of the following scenario: A list of imaging opportunities is presented to the operator after his or her request for imaging of a certain geographical area. Satellite maneuvers, telecommand and telemetry contacts using the connected ground station network are calculated, telecommands are generated and radiated to the spacecraft and their execution is monitored. Imaging data is downlinked, processed and presented to the operator. The whole workflow is performed on the single-laptop hardware and thus serves as demonstration of feasibility and compactness. As next step we show the mobility of the system by taking it off-site for performing the same workflow again, this time without the backing GSOC infrastructure.

The first demonstration is concerned with verifying the compactness of the system: The demonstration operations team performs above workflow from inside GSOC, using the existing network and ground station connections. For this demonstration the regular BIROS Mission Operations System is replaced with the new compact system. The second demonstration is concerned with verifying the mobility of the system: The demonstration operations team performs above workflow from outside GSOC without being able to resort to its infrastructure or facilities. The current candidate for the remote demonstration site is an antenna at Weilheim ground station.

2. Ground Segment Overview

2.1 System Overview

As baseline for the ground segment and operations concept the BIROS project is used due to the BIROS satellite being the spacecraft targeted for the demonstration campaigns. This allows reuse of existing components with minimal adaptions to project-specifics as well as familiarity of the operators running the demonstration campaign. Because the existing BIROS system is still in place it is ensured that operations of the satellite can be resumed from this system at any time, especially during spacecraft or ground contingencies.

The system design shown in Figure 1 is a starting point which evolves in accordance with the project's agile approach. Starting with the demonstrations a potentially usable Mission Operations System is available at any point in time. The following sections detail the composition of each subsystem and how they interface with each other. The concrete system deployment is described in Section 3.

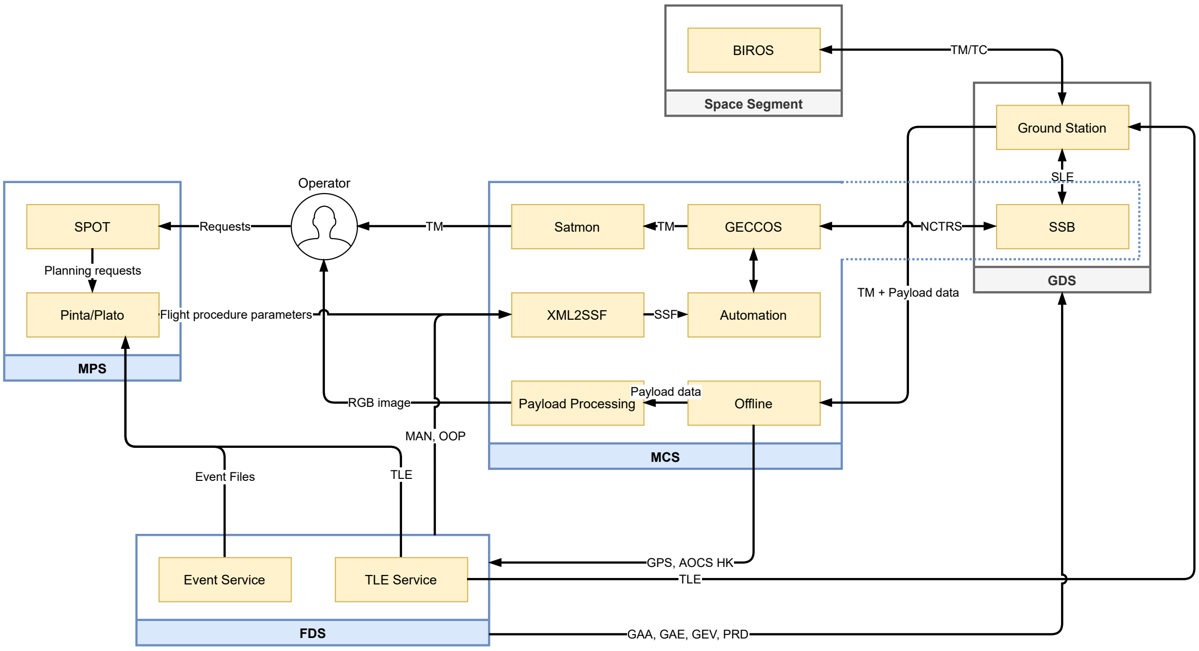


Fig. 1 Overview of the ground segment system design.

Development of the space segment or ground stations (GDS) is not part of this project. The first demonstration campaign uses the new system as plug-in replacement for the existing Mission Operations System. Therefore, some of the existing interfaces have to be served (see Section 2.5 for details). With respect to the overall ground segment

[1] polar sun-synchronous orbit, altitude 510 km, mass ≈ 130 kg, infrared and optical imaging payloads

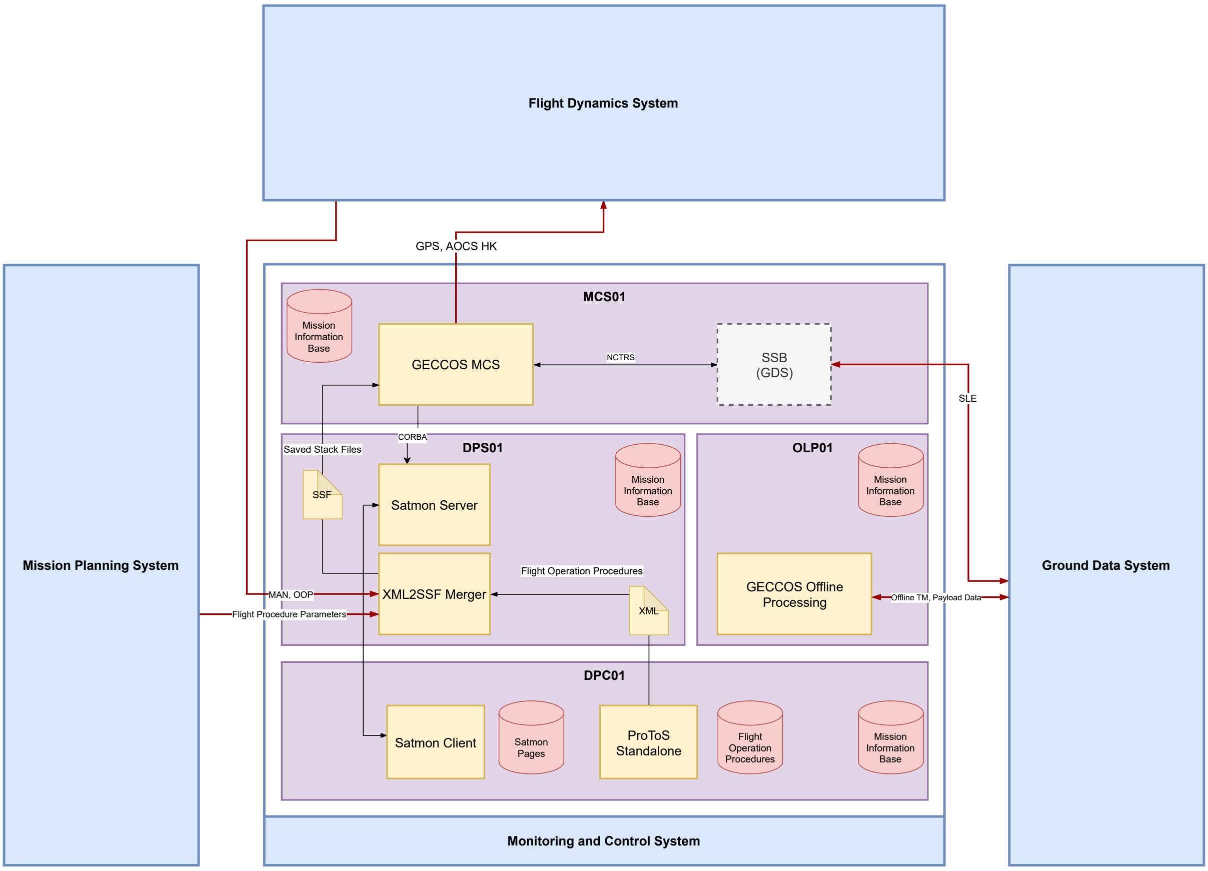


Fig. 2 Overview of the Monitoring and Control System.

design this means that a bespoke protocol (NCTRS) is used for ground station connectivity, which will later be replaced with a standards-compliant protocol (SLE) by integrating one of the GDS components (labeled SSB in Figure 1) into the system.

2.2 Monitoring and Control System

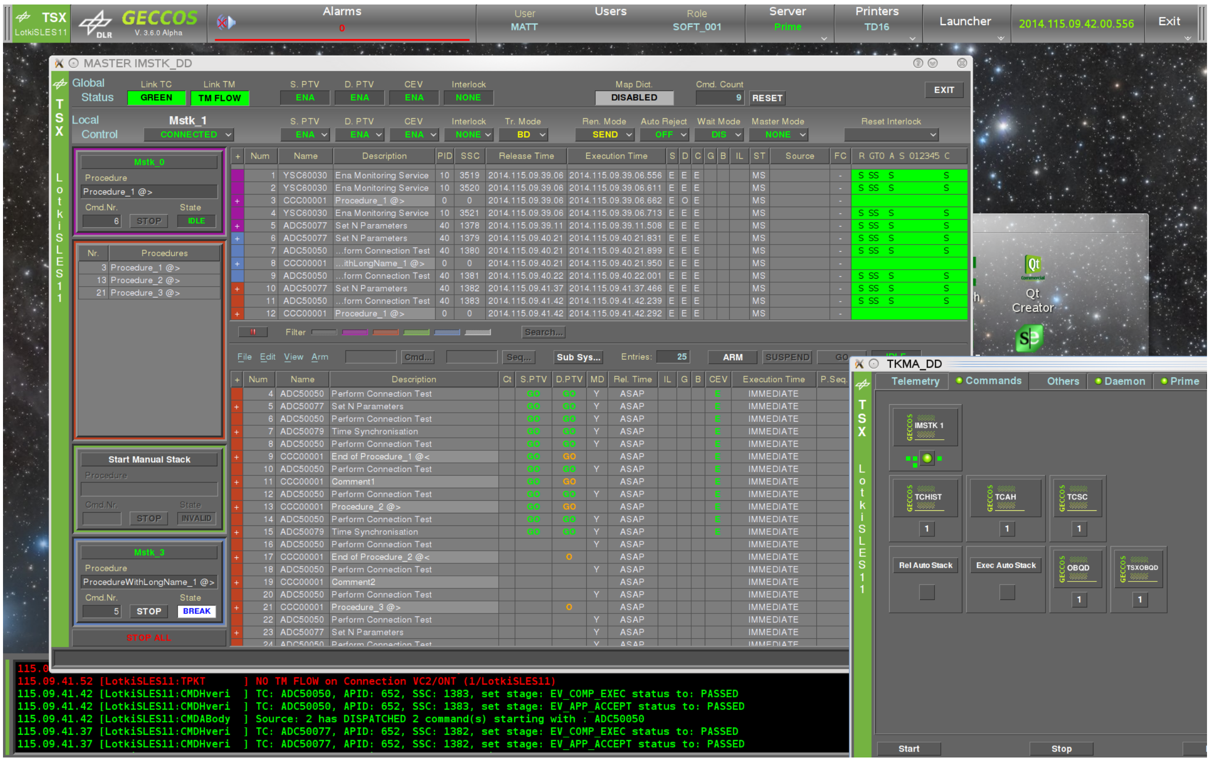
The Monitoring Control System (MCS) is the subsystem that enables the collection, interpretation and archival of satellite telemetry (TM) as well as the preparation and release of telecommands (TC). It is closely coupled with other subsystems like the Ground Data System which forwards telemetry and telecommands to the ground stations and the Flight Dynamics Systems for which it extracts data necessary for orbit calculations. These orbit calculations are then in turn used for scheduling activities by the Mission Planning System. The Monitoring and Control System consists of the following components:

- GECCOS as real-time Monitoring and Control software and recorded TM (i. e. non-real time, "offline") processing system,
- Satmon server and client as display system,
- ProToS as procedure creation, instantiation and automation tool,
- XML2SSF Merger for merging flight procedures with parameter values from several sources,
- Mission Information Base as the common source for TM and TC definitions.

Figure 2 shows a schematic overview of the Monitoring and Control system and its interfaces. Interfaces to subsystems external to MCS are shown with red arrows. The SSB component is part of the Ground Data System and is located there for the first demonstration. It will be moved into the MCS as shown for the second demonstration, thereby paving the way to a standards-compliant antenna interface.

2.2.1 GECCOS

GECCOS [3] is GSOC's custom Monitoring and Control software and has been branched off from ESA's SCOS 2000 which is widely used by ESA for the monitoring and control of ECSS-E-70-41 [4] compatible missions. GECCOS is capable to read telemetry and send telecommands using the bespoke NCTRS interface or can ingest



Global Status — Link TC: GREEN; Link TM: TM FLOW; S. PTV: ENA; D. PTV: ENA; CEV: ENA; Interlock: NONE; Map Dict.: DISABLED; Cmd. Count: 9; RESET; EXIT

Local Control — Mstk_1: CONNECTED; S. PTV: ENA; D. PTV: ENA; CEV: ENA; Interlock: NONE; Tr. Mode: BD; Ren. Mode: SEND; Auto Reject: OFF; Wait Mode: DIS; Master Mode: NONE; Reset Interlock

Mstk_0 — Procedure: Procedure_1 @>; Cmd.Nr.: 6; STOP; State: IDLE

Nr.	Procedures
3	Procedure_1 @>
13	Procedure_2 @>
21	Procedure_3 @>

Start Manual Stack — Procedure; Cmd.Nr.; STOP; State: INVALID

Mstk_3 — Procedure: ProcedureWithLongName_1 @>; Cmd.Nr.: 5; STOP; State: BREAK

STOP ALL

+	Num	Name	Description	PID	SSC	Release Time	Execution Time	S	D	C	G	B	IL	ST	Source	FC	R GT0 A S 012345 C
	1	YSC60030	Ena Monitoring Service	10	3519	2014.115.09.39.06	2014.115.09.39.06.556	E	E	E				MS		-	S SS S S
	2	YSC60030	Ena Monitoring Service	10	3520	2014.115.09.39.06	2014.115.09.39.06.611	E	E	E				MS		-	S SS S S
+	3	CCC00001	Procedure_1 @>	0	0	2014.115.09.39.06	2014.115.09.39.06.662	E	O	E				MS		-	
	4	YSC60030	Ena Monitoring Service	10	3521	2014.115.09.39.06	2014.115.09.39.06.713	E	E	E				MS		-	S SS S S
+	5	ADC50077	Set N Parameters	40	1378	2014.115.09.39.11	2014.115.09.39.11.508	E	E	E				MS		-	S SS S S
+	6	ADC50077	Set N Parameters	40	1379	2014.115.09.40.21	2014.115.09.40.21.831	E	E	E				MS		-	S SS S S
	7	ADC50050	...form Connection Test	40	1380	2014.115.09.40.21	2014.115.09.40.21.899	E	E	E				MS		-	S SS S S
+	8	CCC00001	...ithLongName_1 @>	0	0	2014.115.09.40.21	2014.115.09.40.21.950	E	E	E				MS		-	
	9	ADC50050	...form Connection Test	40	1381	2014.115.09.40.22	2014.115.09.40.22.001	E	E	E				MS		-	S SS S S
+	10	ADC50077	Set N Parameters	40	1382	2014.115.09.41.37	2014.115.09.41.37.466	E	E	E				MS		-	S SS S S
	11	ADC50050	...form Connection Test	40	1383	2014.115.09.41.42	2014.115.09.41.42.239	E	E	E				MS		-	S SS S S
+	12	CCC00001	Procedure_1 @>	0	0	2014.115.09.41.42	2014.115.09.41.42.292	E	E	E				MS		-	

Filter — Search...

File Edit View Arm — Cmd... Seq... Sub Sys... Entries: 25 — ARM SUSPEND GO

+	Num	Name	Description	Ct	S.PTV	D.PTV	MD	Rel. Time	IL	G	B	CEV	Execution Time	P.Seq.
	4	ADC50050	Perform Connection Test		GO	GO	Y	ASAP				E	IMMEDIATE	
+	5	ADC50077	Set N Parameters		GO	GO	Y	ASAP				E	IMMEDIATE	
	6	ADC50050	Perform Connection Test		GO	GO	Y	ASAP				E	IMMEDIATE	
+	7	ADC50079	Time Synchronisation		GO	GO	Y	ASAP				E	IMMEDIATE	
	8	ADC50050	Perform Connection Test		GO	GO	Y	ASAP				E	IMMEDIATE	
+	9	CCC00001	End of Procedure_1 @<		GO	GO		ASAP				E	IMMEDIATE	
	10	ADC50050	Perform Connection Test		GO	GO	Y	ASAP				E	IMMEDIATE	
+	11	CCC00001	Comment1		GO	GO		ASAP				E	IMMEDIATE	
	12	ADC50050	Perform Connection Test		GO	GO	Y	ASAP				E	IMMEDIATE	
+	13	CCC00001	Procedure_2 @>		GO	GO		ASAP				E	IMMEDIATE	
	14	ADC50050	Perform Connection Test		GO	GO	Y	ASAP				E	IMMEDIATE	
+	15	ADC50079	Time Synchronisation		GO	GO	Y	ASAP				E	IMMEDIATE	
	16	ADC50050	Perform Connection Test				Y	ASAP					IMMEDIATE	
+	17	CCC00001	End of Procedure_2 @<			O		ASAP					IMMEDIATE	
	18	ADC50050	Perform Connection Test				Y	ASAP					IMMEDIATE	
+	19	CCC00001	Comment2					ASAP					IMMEDIATE	
	20	ADC50050	Perform Connection Test				Y	ASAP					IMMEDIATE	
+	21	CCC00001	Procedure_3 @>			O		ASAP					IMMEDIATE	
	22	ADC50050	Perform Connection Test				Y	ASAP					IMMEDIATE	
+	23	ADC50077	Set N Parameters				Y	ASAP					IMMEDIATE	
	24	ADC50050	Perform Connection Test				Y	ASAP					IMMEDIATE	

```
115.0
115.09.41.52 [LotkiSLES11:TPKT      ] NO TM FLOW on Connection VC2/ONT (1/LotkiSLES11)
115.09.41.42 [LotkiSLES11:CMDHveri  ] TC: ADC50050, APID: 652, SSC: 1383, set stage: EV_COMP_EXEC status to: PASSED
115.09.41.42 [LotkiSLES11:CMDHveri  ] TC: ADC50050, APID: 652, SSC: 1383, set stage: EV_APP_ACCEPT status to: PASSED
115.09.41.42 [LotkiSLES11:CMDABody  ] Source: 2 has DISPATCHED 2 command(s) starting with : ADC50050
115.09.41.37 [LotkiSLES11:CMDHveri  ] TC: ADC50077, APID: 652, SSC: 1382, set stage: EV_COMP_EXEC status to: PASSED
115.09.41.37 [LotkiSLES11:CMDHveri  ] TC: ADC50077, APID: 652, SSC: 1382, set stage: EV_APP_ACCEPT status to: PASSED
```

Fig. 3 GECCOS screenshot showing the Manual Stack.

telemetry from files. It therefore serves as both, a real-time MCS and an Offline Processing System. The bespoke NCTRS protocol is translated to the standard SLE protocol for communication with ground stations by means of a component labeled SSB (SLE Switch Board). As shown in Figure 2 GECCOS interfaces with nearly all other MCS components and several other subsystems. Figure 3 shows the Manual Stack, which is one of various sub-applications of GECCOS responsible for the preparation and release of telecommands.

2.2.2 *Satmon*

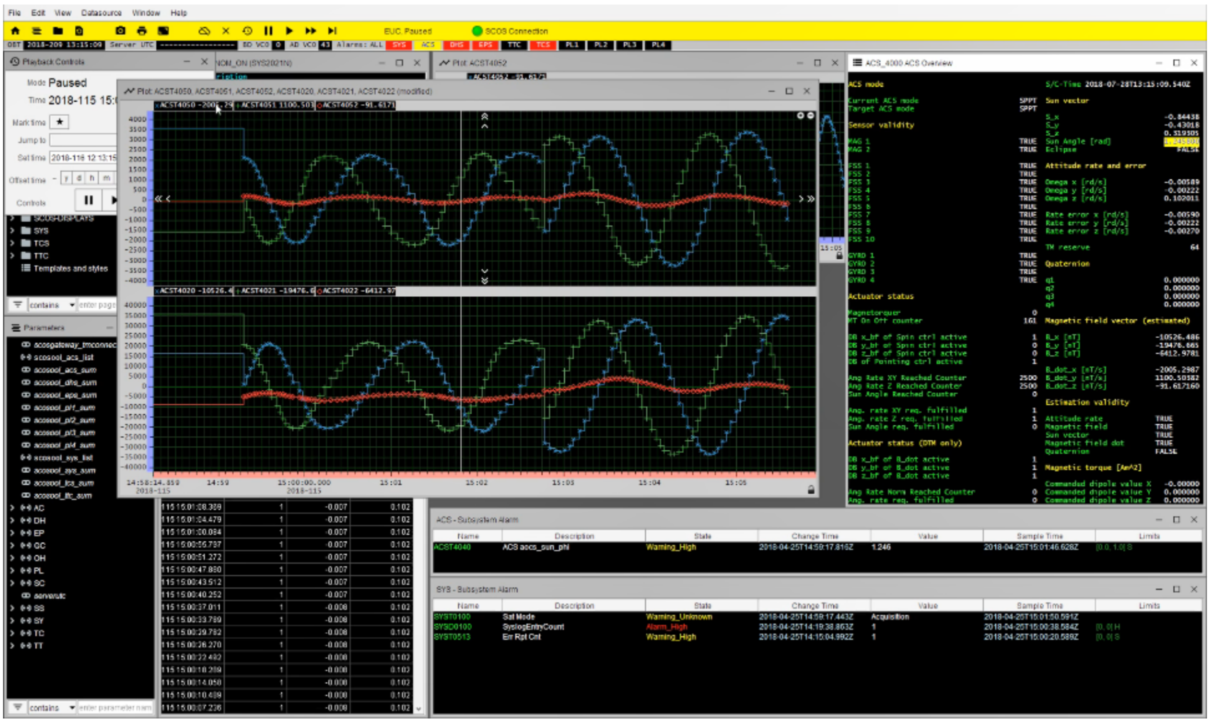
Satmon [5] is the main tool used for the visualization of telemetry parameters. The Satmon client is able to display the incoming telemetry to users in real-time and provides fast access to the history of received parameters. For this task it provides many tools to display telemetry such as lists, aggregated parameter pages, purpose-built overview pages, procedure pages, interactive plots as well as reactive flow charts. An integrated editor enables the user to customize and create telemetry overviews. Figure 4 shows some of the possible displays. A highly efficient telemetry database optimized for high storage density and low retrieval latency backs the Satmon client on the server side.

2.2.3 *ProToS*

The Procedure Tool Suite (ProToS) [6] is a software solution developed at GSOC. Its purpose is to support the creation and execution of satellite test and flight operations procedures and to provide an automation framework for complex operational scenarios. A screenshot of the tool is depicted in Figure 5.

For the demonstration campaigns ProToS accesses the flight operations procedure database of the BIROS flight operations system and enables the operator to instantiate editable command parameters of these procedures before handing them off to the XML2SSF Merger. ProToS is also used for authoring and validating new procedures supporting the demonstration scenarios.

Instantiated procedures are executed manually by the operator for the demonstration campaigns, but it is planned to leverage both ProToS' execution and automation framework later in the project to better support workflow-driven operations: The ProToS execution engine connects directly to GECCOS and supervises procedure execution. This enables execution of branching procedures, which is not possible in an automated way when using Saved Stack Files—the native GECCOS format. The scriptable automation engine can react to different events and trigger execution of procedures as needed [7]. The engineering of the automation requirements and their implementation is future work during the course of this project.



File Edit View Datasource Window Help
EUC Paused
SCOS Connection
Playback Controls
Mode Paused
Plot: ACST4050, ACST4051, ACST4052, ACST4020, ACST4021, ACST4022 (modified)
Plot: ACST4052
ACS_4000 ACS Overview
Actuator status (DTM only)
Magnetic torque [Am^2]
Estimation validity

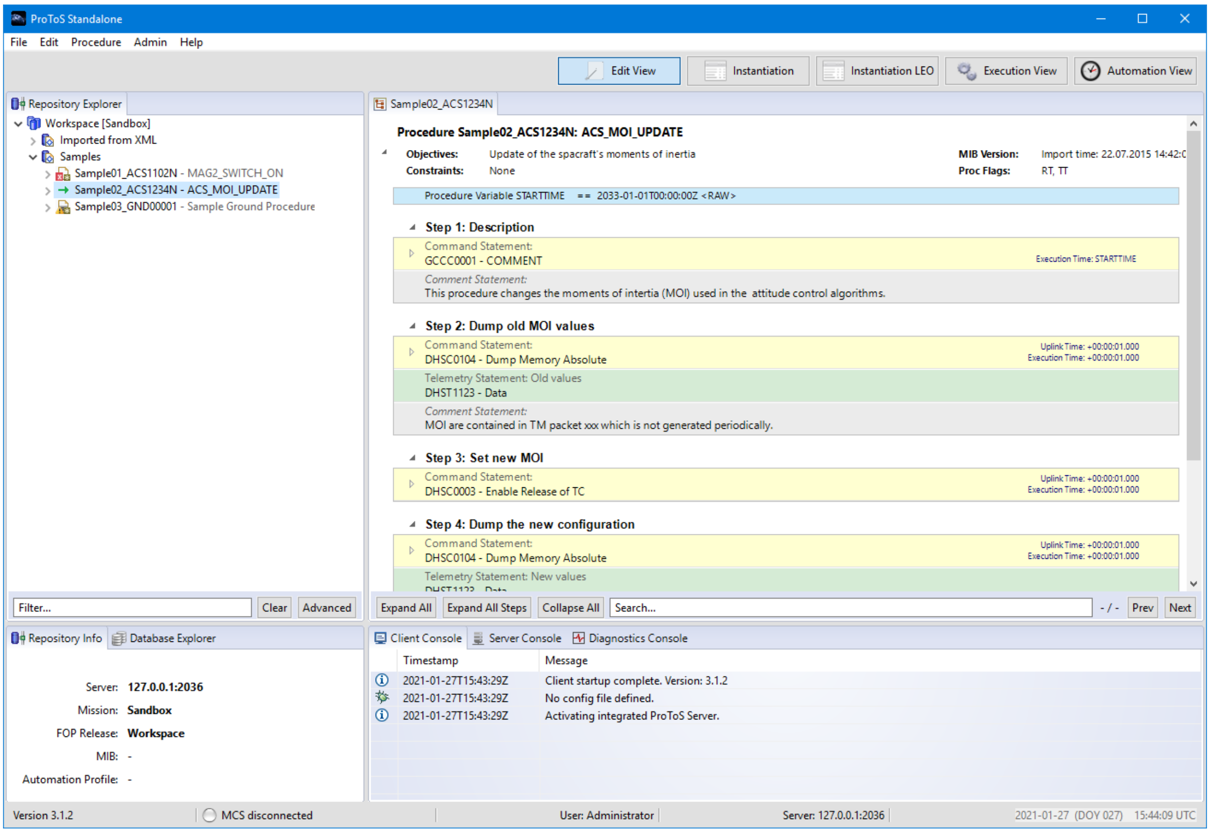
ACS - Subsystem Alarm

Name	Description	State	Change Time	Value	Sample Time	Limits
ACST4040	ACS aocs_sun_phi	Warning_High	2018-04-25T14:59:17.816Z	1.246	2018-04-25T15:01:46.628Z	[0.0, 1.0] S

SYS - Subsystem Alarm

Name	Description	State	Change Time	Value	Sample Time	Limits
SYST0100	Sat Mode	Warning_Unknown	2018-04-25T14:59:17.443Z	Acquisition	2018-04-25T15:01:50.591Z	
SYSD0100	SyslogEntryCount	Alarm_High	2018-04-25T14:19:38.863Z	1	2018-04-25T15:00:38.584Z	[0, 0] H
SYST0513	Err Rpt Cnt	Warning_High	2018-04-25T14:15:04.992Z	1	2018-04-25T15:00:20.589Z	[0, 0] S

Fig. 4 Satmon screenshot demonstrating different telemetry views and data plots.



Workspace [Sandbox]
- Imported from XML
- Samples
  - Sample01_ACS1102N - MAG2_SWITCH_ON
  - Sample02_ACS1234N - ACS_MOI_UPDATE
  - Sample03_GND00001 - Sample Ground Procedure

Sample02_ACS1234N

Procedure Sample02_ACS1234N: ACS_MOI_UPDATE

Objectives: Update of the spacraft's moments of inertia
Constraints: None
MIB Version: Import time: 22.07.2015 14:42:0
Proc Flags: RT, TT

Procedure Variable STARTTIME == 2033-01-01T00:00:00Z <RAW>

Step 1: Description

Command Statement:
GCCC0001 - COMMENT
Execution Time: STARTTIME

Comment Statement:
This procedure changes the moments of intertia (MOI) used in the attitude control algorithms.

Step 2: Dump old MOI values

Command Statement:
DHSC0104 - Dump Memory Absolute
Uplink Time: +00:00:01.000
Execution Time: +00:00:01.000

Telemetry Statement: Old values
DHST1123 - Data

Comment Statement:
MOI are contained in TM packet xxx which is not generated periodically.

Step 3: Set new MOI

Command Statement:
DHSC0003 - Enable Release of TC
Uplink Time: +00:00:01.000
Execution Time: +00:00:01.000

Step 4: Dump the new configuration

Command Statement:
DHSC0104 - Dump Memory Absolute
Uplink Time: +00:00:01.000
Execution Time: +00:00:01.000

Telemetry Statement: New values
DHST1123 - Data

Server: 127.0.0.1:2036
Mission: Sandbox
FOP Release: Workspace
MIB: -
Automation Profile: -

Timestamp	Message
2021-01-27T15:43:29Z	Client startup complete. Version: 3.1.2
2021-01-27T15:43:29Z	No config file defined.
2021-01-27T15:43:29Z	Activating integrated ProToS Server.

Version 3.1.2 | MCS disconnected | User: Administrator | Server: 127.0.0.1:2036 | 2021-01-27 (DOY 027) 15:44:09 UTC

Fig. 5 Screenshot of ProToS showing the tabular view of an example flight procedure.

2.2.4 Other Components

XML2SSF Merger The XML2SSF Merger merges flight procedure templates in XML format with parameter values in order to provide instantiated flight procedures in a format suitable for GECCOS (SSF). Parameter values are ingested from operators via ProToS, from the Mission Planning timeline, or from Flight Dynamics maneuver activities.

Mission Information Base The Mission Information Base (MIB) is used by all of the tools above and is the central source for telemetry and telecommand definitions.

SLE Switchboard The SLE Switchboard (SSB) is a software to support the interface with various types of ground station equipment. It acts as a protocol bridge between the GECCOS-bespoke NCTRS protocol and the standard SLE protocol supported by ground stations throughout the world supporting acquisition of telemetry and sending of telecommands.

2.3 Mission Planning System

The main purpose of the Mission Planning System (MPS) in context of this project is the generation of consistent, conflict-free timelines and sequences of flight operations procedures (FOPs) in order to command the payload and background sequence operations of the target spacecraft from all given input items and known constraints. In addition, MPS shall support the operator in the pre-planning and ordering process for acquisitions of the spacecraft imaging payload.

The main MPS task for the demonstrations is to adapt the already existing MPS components used for BIROS in the scope of the FireBird mission [8] to the needs of the project. The main challenges here are the lack of a connection to the Internet, as available during a regular scientific mission like FireBird, and the usage of a different payload for the demonstration campaigns than the one used during the FireBird mission. The primary payload for the FireBird mission is a scanline infrared imaging device, whereas the demonstrations use the secondary payload optical camera that yields area images. Figure 6 gives an overview of the MPS components and interfaces and how they interact with the operator and the Flight Dynamics and Monitoring & Control subsystems. It can be seen that the MPS comprises of two components which will be further described in the following. Scheduling can either be an external component or integrated into MPS.

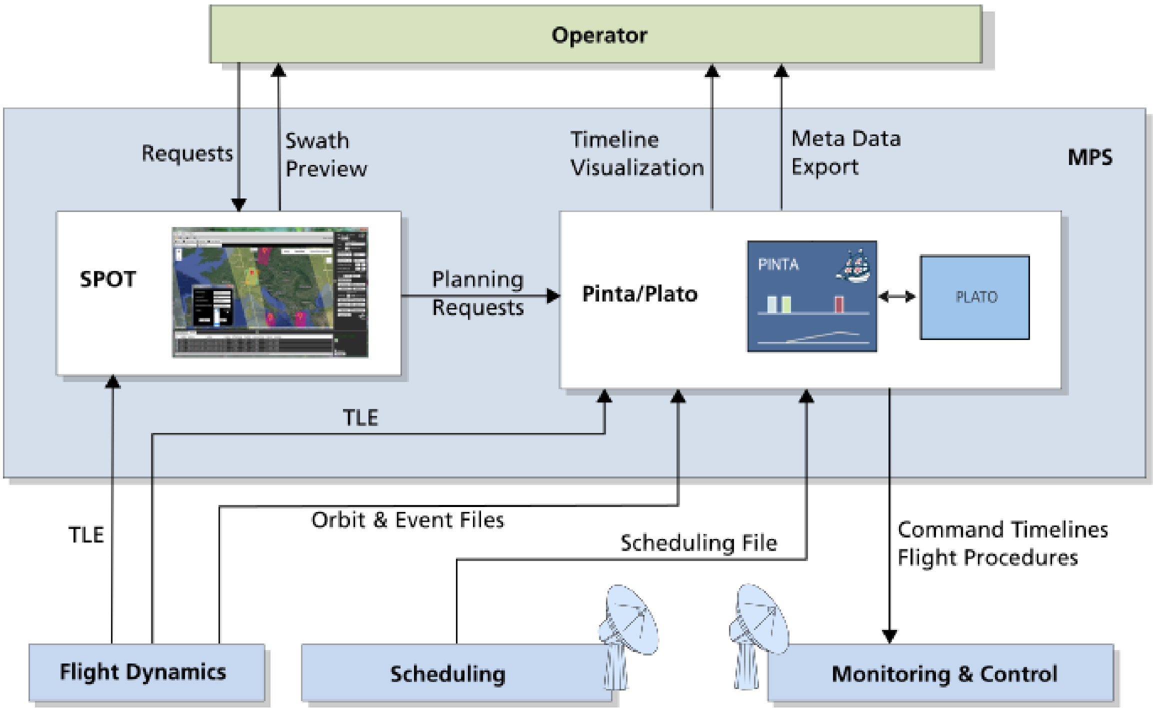
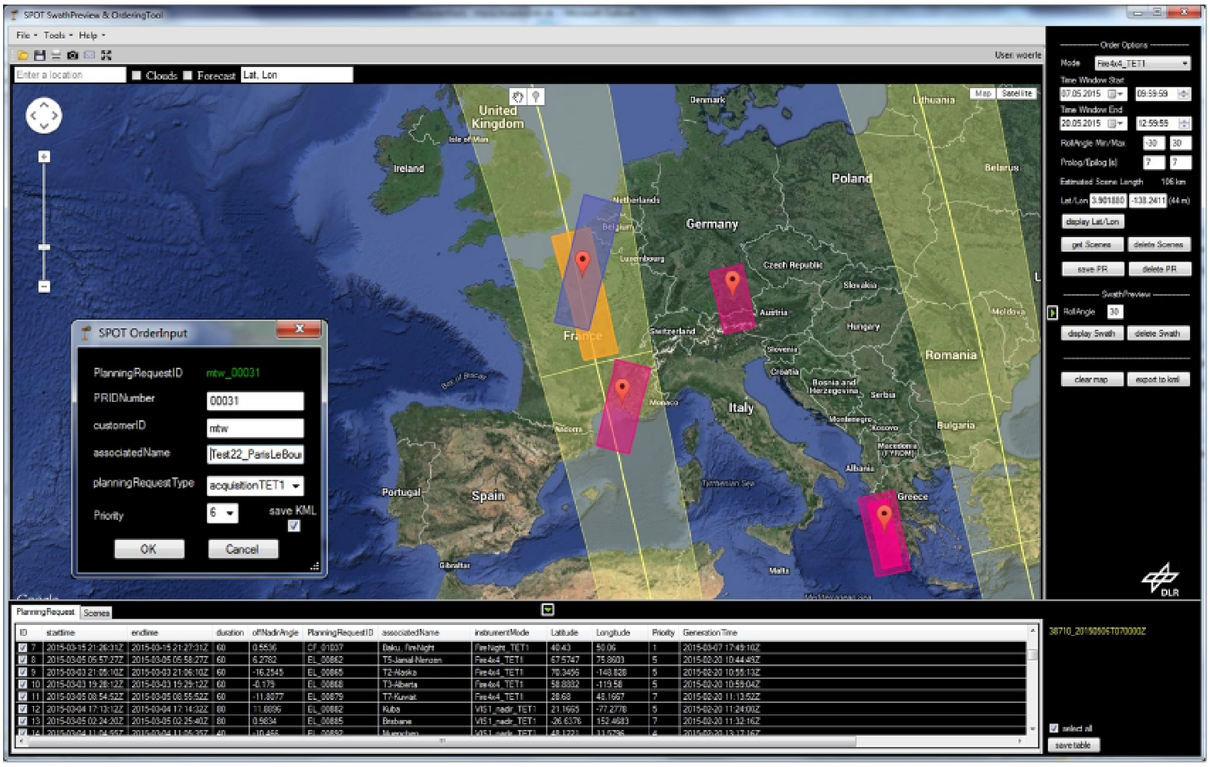


Fig. 6 Schematic view of the Mission Planning System components and interfaces.



SPOT OrderInput

PlanningRequestID mtw_00031
PRIDNumber 00031
customerID mtw
associatedName Test22_ParisLeBou
planningRequestType acquisitionTET1 ▾
Priority 6 ▾
save KML ☑

OK Cancel

Order Options
Mode Fire4x4_TET1 ▾
Time Window Start 07.05.2015 09:59:59
Time Window End 20.05.2015 12:59:59
RollAngle Min/Max -30 30
Prolog/Epilog [s] 7 7
Estimated Scene Length 106 km
Lat/Lon 3.901880 -138.2411 (44 m)

display Lat/Lon
get Scenes delete Scenes
save PR delete PR

SwathPreview
RollAngle 30

display Swath delete Swath
clear map export to kml

PlanningRequest Scenes

ID	starttime	endtime	duration	offNadirAngle	PlanningRequestID	associatedName	instrumentMode	Latitude	Longitude	Priority	Generation Time
☑ 7	2015-03-15 21:26:31Z	2015-03-15 21:27:31Z	60	0.5536	CF_01037	Baku_FireNight	FireNight_TET1	40.43	50.06	1	2015-03-07 17:49:10Z
☑ 8	2015-03-05 05:57:27Z	2015-03-05 05:58:27Z	60	6.2782	EL_00862	T5-Jamal-Nenzen	Fire4x4_TET1	67.5747	75.8603	5	2015-02-20 10:44:49Z
☑ 9	2015-03-03 21:05:10Z	2015-03-03 21:06:10Z	60	-16.2545	EL_00865	T2-Alaska	Fire4x4_TET1	70.3456	-148.828	5	2015-02-20 10:55:13Z
☑ 10	2015-03-03 19:28:12Z	2015-03-03 19:29:12Z	60	-0.179	EL_00868	T3-Alberta	Fire4x4_TET1	58.8832	-119.58	5	2015-02-20 10:59:04Z
☑ 11	2015-03-05 08:54:52Z	2015-03-05 08:55:52Z	60	-11.8077	EL_00879	T7-Kuwait	Fire4x4_TET1	28.68	48.1667	7	2015-02-20 11:13:52Z
☑ 12	2015-03-04 17:13:12Z	2015-03-04 17:14:32Z	80	11.8896	EL_00882	Kuba	VIS1_nadir_TET1	21.1665	-77.2778	5	2015-02-20 11:24:00Z
☑ 13	2015-03-05 02:24:20Z	2015-03-05 02:25:40Z	80	0.5834	EL_00885	Brisbane	VIS1_nadir_TET1	-26.6376	152.4683	7	2015-02-20 11:32:16Z
☑ 14	2015-03-04 11:04:55Z	2015-03-04 11:05:35Z	40	-10.466	EL_00892	Muenchen	VIS1_nadir_TET1	48.1221	11.5796	4	2015-02-20 13:17:16Z

38710_20150506T070000Z

☑ select all
save table

Fig. 7 Snapshot of the Swath Preview and Ordering Tool (SPOT).

2.3.1 *Swath Preview and Ordering Tool (SPOT)*

The GSOC Swath Preview and Ordering Tool (SPOT) is used for the calculation and visualization of upcoming acquisition opportunities for areas of interest and for the generation of consistent planning requests for the spacecraft controlled by this Mission Operations System on the one hand and the visualization of the swaths of the spacecraft for the pre-planning of campaigns on the other hand.

SPOT provides a graphical user interface using GSOC's SCOTA library and embedding Google Maps[1] or OpenStreetMap[2]. The maps are replaced with an offline solution for this project due to the potential lack of Internet connectivity. SPOT allows to calculate target visibilities based on the latest two-line elements from Flight Dynamics and to prepare consistent planning requests from the chosen target acquisition opportunities, which are to be sent to the core planning system and contain all necessary planning information. A snapshot of the current version and layout of the SPOT GUI can be seen in Figure 7.

2.3.2 *Pinta/Plato*

The planning runs for the project are performed with this tool for semi-automated planning & scheduling and timeline export. It is based on GSOC's generic Program for Interactive Timeline Analysis (PINTA) and the GSOC PLATO library and maintains the current planning model based on the latest inputs with project-specific, configurable algorithms and plug-ins. At the beginning of every planning run, Pinta/Plato collects all relevant information:

- the planning requests from the operator that have been created via SPOT,
- the information about scheduled ground station contacts—obtained through a central scheduling office if available or through automated or interactive user selection,
- the orbit-related information from Flight Dynamics, containing e. g. the timestamps of shadow events and ground station visibilities,
- the current two-line elements from Flight Dynamics.

By applying a dedicated combination of configurable planning algorithms, Pinta/Plato enables scheduling timeline entries of all necessary flight operations procedures to run the payload operations and background sequence tasks of the mission and generating conflict-free timelines. The execution timeline consisting of a sequence of flight operation procedures and their command parameters are handed over to the XML2SSF Merger. The operator has

[1] https://maps.google.com/

[2] https://www.openstreetmap.org/

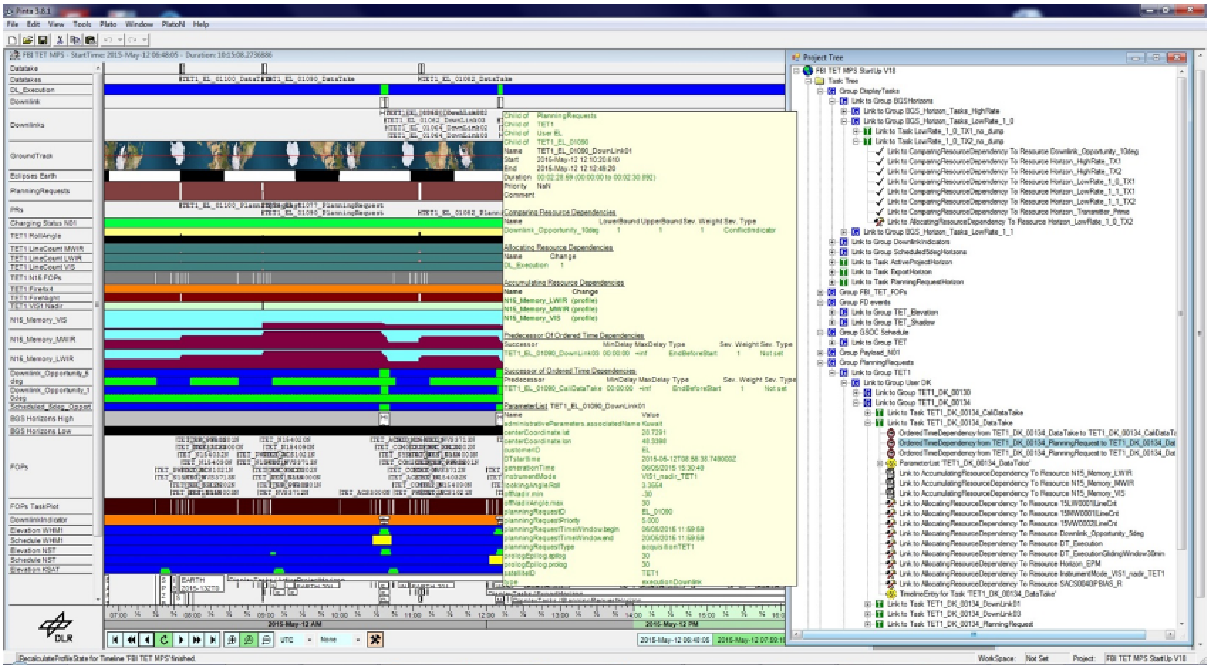


Fig. 8 Snapshot of the Program for Interactive Timeline Analysis (PINTA), combined with the Planning Tool (PLATO) as well as generic and project-specific plug-ins.

the possibility to check and modify the results of a planning run by viewing the graphical timeline representation provided by PINTA. A snapshot of the current version of the timeline view can be seen in Figure 8.

2.4 Flight Dynamics System

The Flight Dynamics System (FDS) is responsible for computing any information related to the satellite orbit position and attitude. A LEO satellite equipped with a GNSS receiver shall serve as baseline in the context of this project. Transponder ranging presents a second source of navigation measurements, that will be used during launch and early orbit phase and as fallback. In this basic form the FDS tasks include tracking data pre-processing and conversion, orbit determination (OD), and generation of orbit-related products like events and two-line elements (TLE) for observing ground stations. The orbit information also serves as essential input for planning of maneuver activities or Earth observation campaigns.

Depending on the specific satellite payload the FDS may further generate additional data products. The satellite payload also drives the requirements for product quality, like the ensured orbit determination and prediction accuracy. In an enclosed network environment the FDS may not receive updates of auxiliary data for Earth rotation parameters and solar flux predictions. In the following, the maximum position errors are analyzed in the case of missing updates of the auxiliary data.

2.4.1 Solar Activity

The density models of the residual atmosphere require as input the solar radiation flux at 10.7 cm wavelength, the 90-day average value, as well as the Kp-index and ap-index of geomagnetic activity. Without current data on solar activity, mean value can be applied as typical values. The estimated drag coefficient will compensate for the average atmospheric drag the satellite encountered during the observation arc. In this way long periodic changes are covered, and in particular the 11-year solar cycle.

To validate this, the daily orbit determination runs of the BIROS satellite have been reprocessed with mean solar flux data. The first four years of BIROS operations coincide with minimum solar activity. For the most part the average flux values applied for orbit determination exceed the actual solar activity. This is compensated by lower estimates of the drag coefficient during this time period of low solar activity. In total 1353 orbit records from the first four years of BIROS operations are determined with constant mean values for solar activity, and further propagated for a prediction time of up to four days. The corresponding orbit records from Mission Operations are propagated for the same time periods and using the daily prediction file on solar activity. The position difference from the 1353 ephemeris pairs allows for a statistical analysis of position errors. As can be seen from Figure 9 the mean position errors in radial, along-track and cross-track direction mostly cancel out for the large number of OD

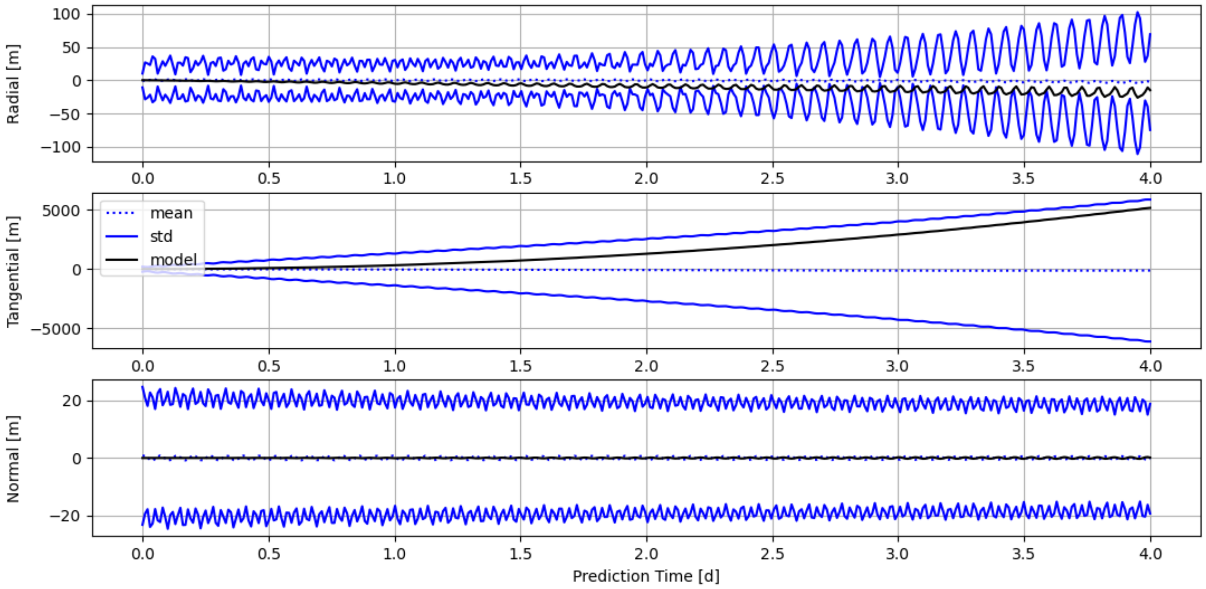


Fig. 9 Position differences from average solar flux data derived from past mission data and error model.

cases. The 1σ error bounds are plotted with solid blue lines. After three days of orbit prediction the 1σ position error is about 4.15 km, mostly in along-track direction. Note that these statistical errors are derived from a period of low solar activity.

The daily prediction files from January 2001 till September 2020 are compared against the timeline of historical flux data. The standard deviation of the difference between forecast and observation is evaluated depending on the prediction time and in time intervals of one day. The standard deviation corresponds to the 1σ error of the solar flux prediction. In 68 % of the cases the actual value is within the 1σ interval, in 95 % within the 2σ interval, in 99.7 % within the 3σ interval, etc. The 2σ bound is applied for estimation of the maximum orbit errors, refer to Table 1.

The expected position errors are to be estimated by means of a circular polar orbit. The orbit is propagated numerically over a period of three days with constant mean values of solar activity, and then over the same period with solar activity set to the highest expected solar activity (MEAN + 2 · STD). A corresponding 2σ position error is computed from the difference of the two orbit predictions, e. g. in 95 % of the cases a lower solar activity is to be expected, and consequently lower position errors, and in 5 % of the cases the solar activity is higher with corresponding higher position errors than indicated. When assuming a higher solar activity the satellite experiences larger atmospheric drag. The position errors in flight direction grow fastest with an exponential increase, followed by the position errors in radial direction. Until the end of the prediction period of three days, there are only small differences in normal direction. The 3-dimensional position error after three days is about 60 km.

Figure 10 illustrates the dependency of the maximum 3-D position error from orbit height. Depending on the required orbit accuracy, orbits created without daily forecasts of solar activity may be used only over a limited prediction time. Besides the strong dependence on altitude, the satellite's area-to-mass ratio and drag coefficient also have an influence. In this example a ratio of 100 kg m^{-2} and a drag coefficient of 1.5 are applied.

Table 1 Solar flux data applied for maximum error estimation (2σ errors).

	MEAN	STD			
		Day 1	Day 2	Day 3	Day 4
F10.7 flux	221.3 sfu	6.7 sfu	8.6 sfu	10.5 sfu	12.3 sfu
Average F10.7 flux	213.8 sfu	0.28 sfu	0.31 sfu	0.37 sfu	0.47 sfu
Kp-Index	4.92	1.45	1.41	1.38	1.38
ap-Index	39.67	10.3	11.1	11.4	11.4

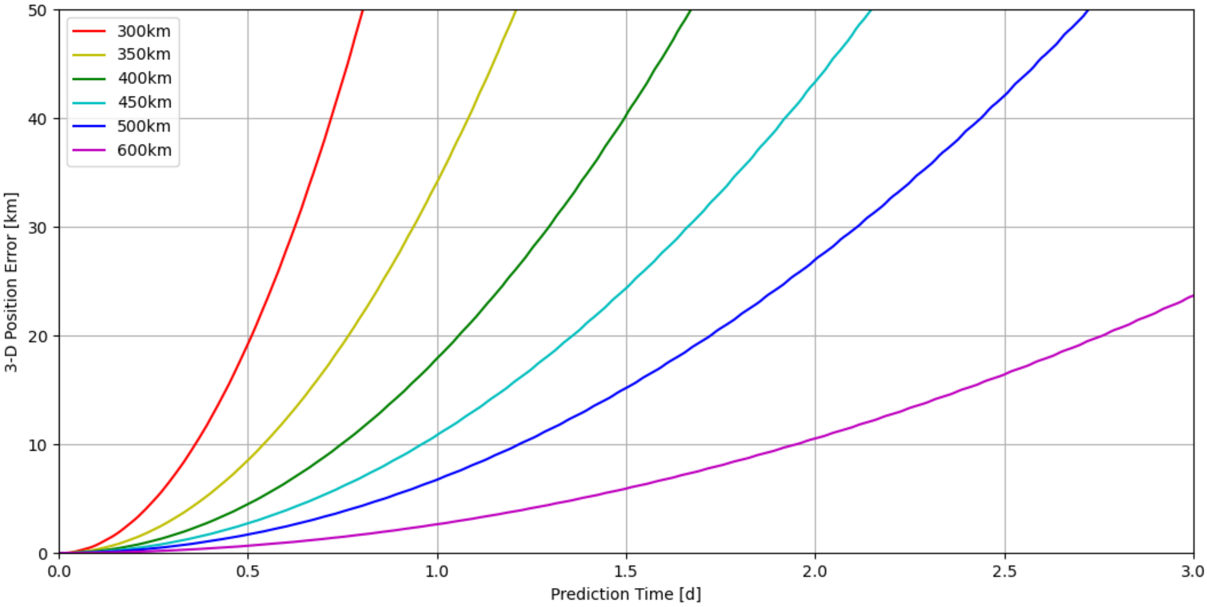


Fig. 10 2σ position errors at various orbit heights, due to missing forecasts of solar activity.

2.4.2 Earth Rotation Parameters and Leap Seconds

Missing updates of the Earth rotation parameters (ERP) lead to a faulty coordinate transformation between Earth-fixed frames and inertial reference frames. The maximum position errors are estimated without consideration of further orbit prediction errors.

The pole coordinates (x-pole, y-pole) describe the precise position of the Earth's rotation axis, which follows a circle with a radius angle of $2.91 \cdot 10^{-6}$ rad. The maximum position error scales with the distance to the center of the Earth. For example, it is about 20 m at an altitude of 500 km.

The time difference UT1 − UTC defines the phase of Earth rotation with respect to the vernal equinox. A time error is translated into an angular error by the Earth rotation rate of $7.292\,115 \cdot 10^{-5}$ rad s^{-1}. A new leap second is typically introduced when the time difference is about +0.5 s or −0.5 s. Thus, the maximum position error shall be estimated for an unconsidered time difference UT1 − UTC of half a second. The corresponding angular error is scaled with the distance to the center of the Earth, e. g. about 250 m for an orbit height of 500 km and a UT1 − UTC error of 0.5 s. The transformation between Earth-fixed and inertial coordinate systems is needed several times. Station coordinates (transponder ranging) or on-board GNSS measurements are defined in Earth-fixed coordinates. Orbit elements are defined in an inertial coordinate system, and numerical orbit propagation for orbit determination and orbit prediction is performed in an inertial system. Therefore, the estimated error applies for all data products derived from inertial coordinate frames: TLE, osculating orbit elements, orbit ephemeris in inertial frame, derived close approach warnings, etc.

If data products are derived from orbit information in Earth-fixed frame, like orbit ephemeris, the transformation errors to and from inertial frame may cancel out to some extent. To demonstrate this, a single orbit determination run for the BIROS satellite from 2017-03-12 is re-run with and without ERP data. At that date the time difference UT1 − UTC was approximately +0.5 s. For the OD run with missing ERP data, the time differences UT1 − UTC and the pole coordinates are set to zero. The observation arc covers 24 h of on-board GPS observations.

The position differences of the two solutions are compared in True-of-Date inertial coordinate frame and ICRF2000 Earth-fixed frame. A good fraction of the transformation errors cancel for the Earth-fixed frame during the observation arc, see Figure 11. The right ascension of the ascending node is shifted by approximately 0.002° for the orbit estimated without ERP data, corresponding to the Earth rotation angle during half a second. The estimated orbit elements are propagated for a further prediction time of up to three days. The position errors in Earth-fixed frame start to grow outside the observation arc. For longer propagation times the position errors exceed the level of the transformation errors, both in Earth-fixed and inertial coordinate frames.

Like errors in the time difference UT1 − UTC, missing leap seconds lead to faulty coordinate transformations, and the position error can be estimated in the same way. If new leap seconds are not equally introduced to all sub-systems of the ground and space segment, relative time differences arise. The position error due to relative time errors results from the orbit velocity, e. g. 7.6 km per leap second at an orbit height of 500 km.

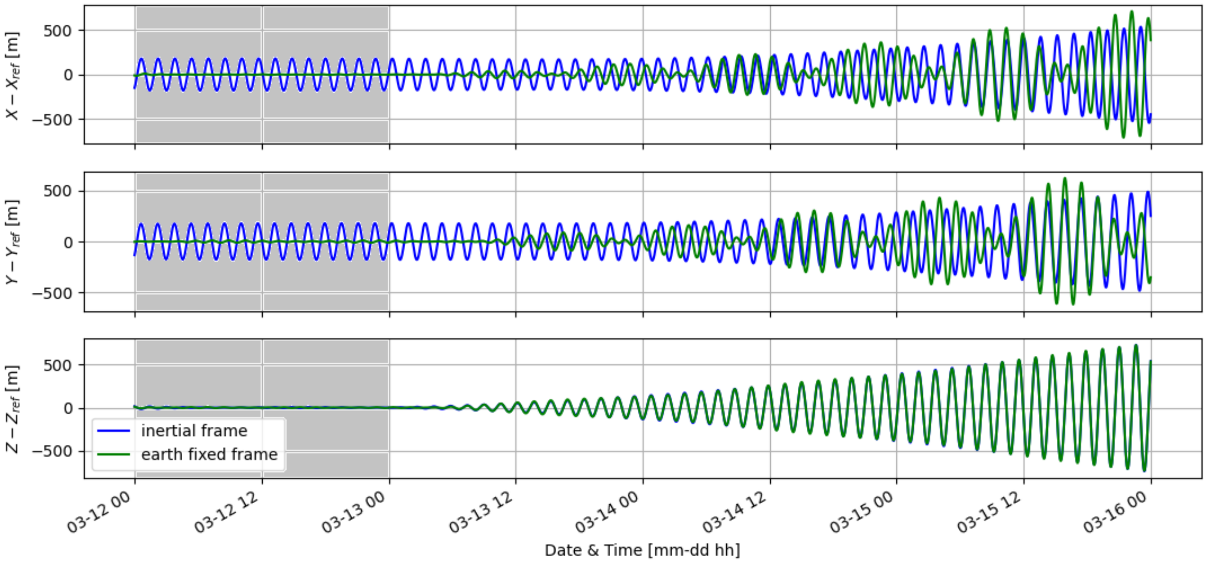


Fig. 11 Position errors due to missing Earth rotation parameter (observation arc in grey).

2.5 External Interfaces

Although the presented ground system aims to be as self-sufficient as possible, any real-world deployment needs to interface with external systems. One external interface that is always necessary is the connection to a ground station or ground station network. Ground station connectivity is achieved by means of a bespoke protocol (NCTRS) for the first demonstration and by a standard protocol (SLE) for the second demonstration. Additionally, the demonstration campaigns mandate further interfaces: Synchronization of certain products, like telecommand history, into the existing BIROS Mission Operations System is necessary in order to ensure the safety of the spacecraft and aid analysis in case mission control is exerted again from the existing BIROS system.

For any upcoming user mission the effect of missing or sporadic auxiliary data for Flight Dynamics due to intermittent Internet connectivity has been studied in Section 2.4. These effects will affect the mission design, leading to compromises between orbit position errors and the degree of self-containedness. Missing solar flux auxiliary data have the largest impact on the possible error, accumulating to 60 km in three days for an orbit height of 500 km in a maximum error estimation. These errors for example translate to deviations in the imaged area for an optical payload or to variations of ground station acquisition times. Up-to-date data are injected into the system at the beginning of each demonstration campaign. Possible errors during the short demonstration operations timeframe, which lasts less than three days, are deemed acceptable.

3. Deployment Concept

The deployment concept for this project, i. e. how the systems comprising the ground segment are rolled out on physical or virtual hardware and how they connect to each other, shall be outlined in this section.

3.1 Hardware

Owing to the mobility of this project all components are integrated into a single mobile hardware, namely a commercial off-the-shelf (COTS) laptop. Its specifications (128 GiB RAM, 8 core CPU, 4 TB SSD) provide enough headroom for flexible adaption to a user mission and for the integration of more automation functionality. Although the demonstration system runs on powerful hardware because it is used as development testbed, deployment of the ground system is not closely tied to this particular piece of hardware: Project components are routinely rolled out on developer's machines without any changes compared to the production environment. This proves beneficial during the SARS-CoV-2 pandemic, where access to physical hardware is limited, also see Section 4. The strict separation between hardware and ground system deployment information, which will be detailed in the next section, will also allow shipping the ground system as USB stick or download image, ready to be rolled out on user hardware on demand. The latter step is enabled by the chosen system design and will be further fleshed out during and after the two planned demonstration campaigns.

The COTS hardware acts as a host for a number of virtual machines, in which the system components are deployed. No component runs on the physical machine directly. This allows full infrastructure control without imposing too strict requirements on the hosting hardware. The physical host machine runs a SuSE Linux[1] operating system and uses Oracle VirtualBox[2] for virtualization. Both are no strict requirements: It is possible to use Microsoft Windows as host system without any changes. Replacing the virtualization solution is possible in principle, but might require some changes.

The system is built without redundancy in mind, neither for single VMs nor for the physical host. This simplifies workflows and lowers hardware requirements significantly but comes with the risk of losing satellite operability due to hardware or software failure. However, the project enables spinning up a completely new ground segment quickly in response to such a situation or have a second system running in parallel as hot redundancy. At this point in time redundancy concepts, especially state synchronization, are not further fleshed out. Yet, the system design and possible deployment options allow novel redundancy concepts not usually found within classical control centers, e. g. ad-hoc physically distributed places of operations via download of the control center or using cloud infrastructure. The demonstration campaigns are set up in a way that in case of failure the existing BIROS Mission Operations System can take over.

3.2 Infrastructure as Code

"Infrastructure as code" is a phrase that typically denotes the notion of defining data center hardware and provisioning with machine-readable files. Although this project only uses a single physical hardware the same technique can be used to describe the virtual infrastructure. The machine-readable definition files are treated like program source code and thus are checked into a version-controlled repository for development. Ad-hoc machine configuration, e. g. by logging in on a machine and performing configuration changes, is only possible during development and even in this case changes are only temporary. A continuous integration lifecycle recreates the whole infrastructure upon code changes, thereby nullifying any manual changes.

The system deployment is realized with three open source tools: HashiCorp Packer[3], HashiCorp Vagrant[4] and Red Hat Ansible[5]. Figure 12 shows how these tools are used in the context of this project for provisioning the whole system. Once the system is rolled out none of these tools are needed any longer. The steps necessary for running a system without these tools need to be identified as part of the work following the demonstration campaigns.

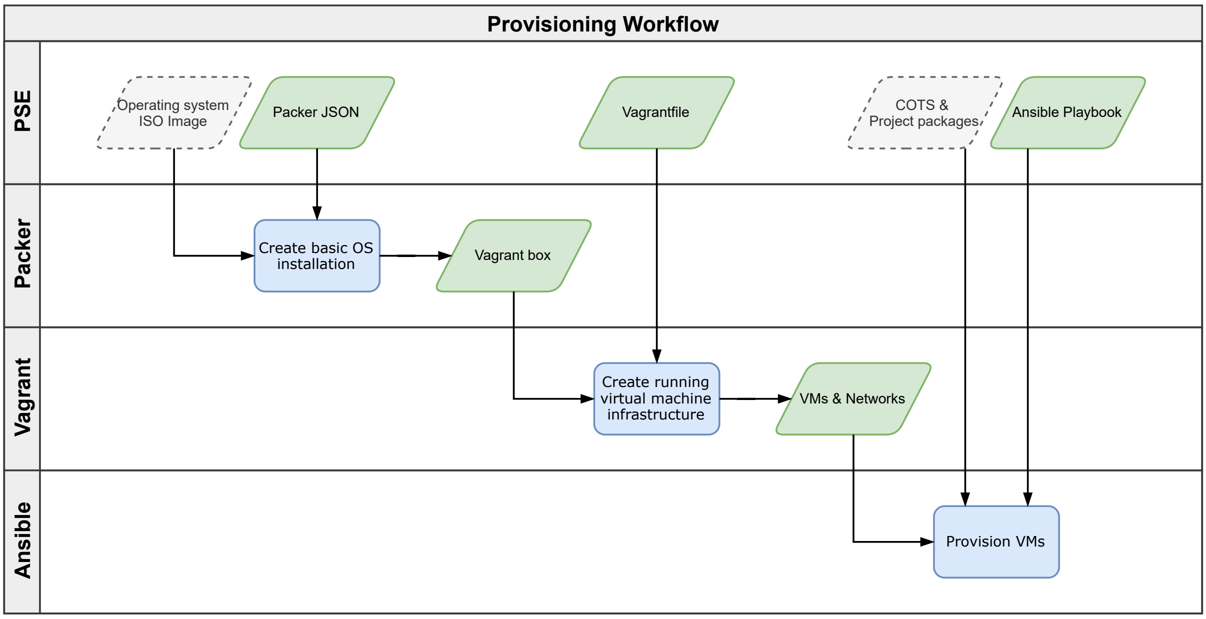


Fig. 12 Provisioning workflow for the roll-out of the ground system. "PSE" is the Project System Engineer.

3.2.1 HashiCorp Packer

Packer is used for creating basic virtual machine images for specific operating systems. A JSON file describes the virtualized hardware and how an operating system is installed into this virtual hardware, starting from the ISO

[1] `https://www.suse.com/`

[2] `https://www.virtualbox.org/`

[3] `https://www.packer.io/`

[4] `https://www.vagrantup.com/`

[5] `https://www.ansible.com/`

images obtained from the operating system manufacturer. The resulting boxes are meant to be generic in order to serve as basis for possibly creating multiple machines from the same box.

3.2.2 HashiCorp Vagrant

Vagrant creates concrete virtual machines by specializing the boxes created in the previous step: It modifies the virtualized hardware resources according to the VM's role, assigns network interfaces and network addresses, mounts storage space (possibly shared between VMs), and triggers execution of the provisioning step with Ansible. It also defines the network rules governing the means by which the VMs may communicate with the outside world, i. e. with other physical machines.

3.2.3 Red Hat Ansible

Ansible works with the virtual machines created in the previous step and puts them in the desired state described by a so-called Playbook file. This involves installing required COTS and specialized project software packages, creating users and groups, configuring the systems and setting up the necessary directories, services and other resources as needed. Ansible is an agentless provisioning system, which means that no software agent is required to run on the machines to be provisioned. Ansible is executed on a control node and connects to each controlled node. For the role of the Ansible control node a dedicated VM was created in the previous step instead of executing Ansible on the physical host machine. This design is chosen in accordance with the policy to keep requirements for the host machine low.

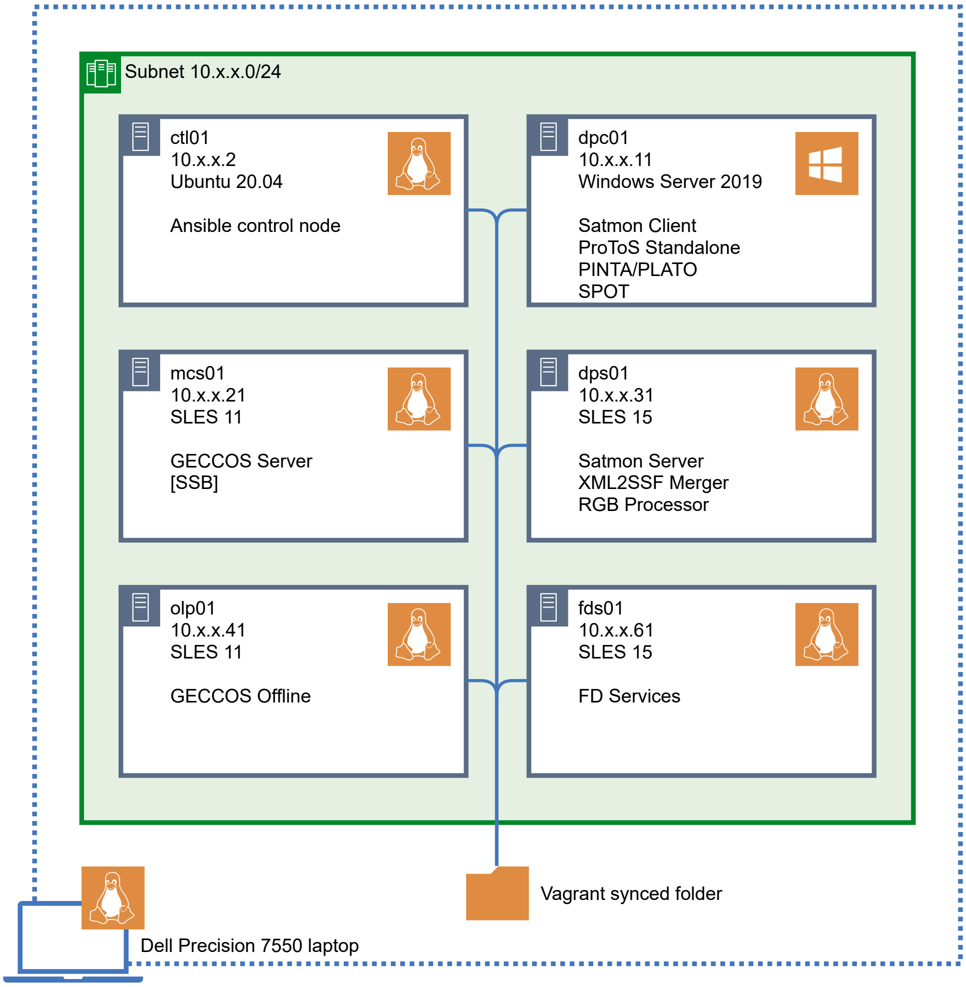


Fig. 13 Ground system infrastructure deployment.

3.3 Virtual Infrastructure Deployment

The infrastructure that is set up using the previously mentioned provisioning tools is shown in Figure 13. All VMs are put in the same network 10.x.x.0/24, which is a private network that allows communication of the VMs with each other. Communication with any outside network is performed by forwarding specific ports from the physical host machine network interface to the appropriate VM. Runtime file data that needs to be persisted, such as telecommand history data, and file data shared between VMs is exchanged via a Vagrant synced folder which is realized as an Oracle VirtualBox shared folder.

3.4 Development Environment

For implementing the ground segment a dedicated development environment has been set up. It runs on the same project hardware as the final product and allows all engineers to develop their systems in parallel. In fact, the final product is merely an artifact of the development environment and itself fully integrated into the same automatic deployment lifecycle as the test systems provided by the development environment. Because all infrastructure is handled as code, typical software engineering tools can be used: The infrastructure definitions and dependencies are checked into a Git[1] repository managed by an in-house GitLab[2] instance. Commits to specific branches trigger different Continuous Integration/Continuous Deployment (CI/CD) pipelines that transform the committed code into running and provisioned infrastructure. Figure 14 shows how this enables automatic creation and removal of several simulation or development environments. Each of these environments represents a complete ground segment and has its own separated network assigned. One of them, the OPS environment, is specially protected and tested more rigorously through a staging environment because it is the environment supposed to be used operationally. The OPS environment can be recreated into a known state directly from the backing infrastructure code at any point in time. All other environments are for the engineers to work with and are rather short-lived. They may be tampered with and also can be easily recreated at any point in time. Changes can only be made persistent by committing them to the Git repository.

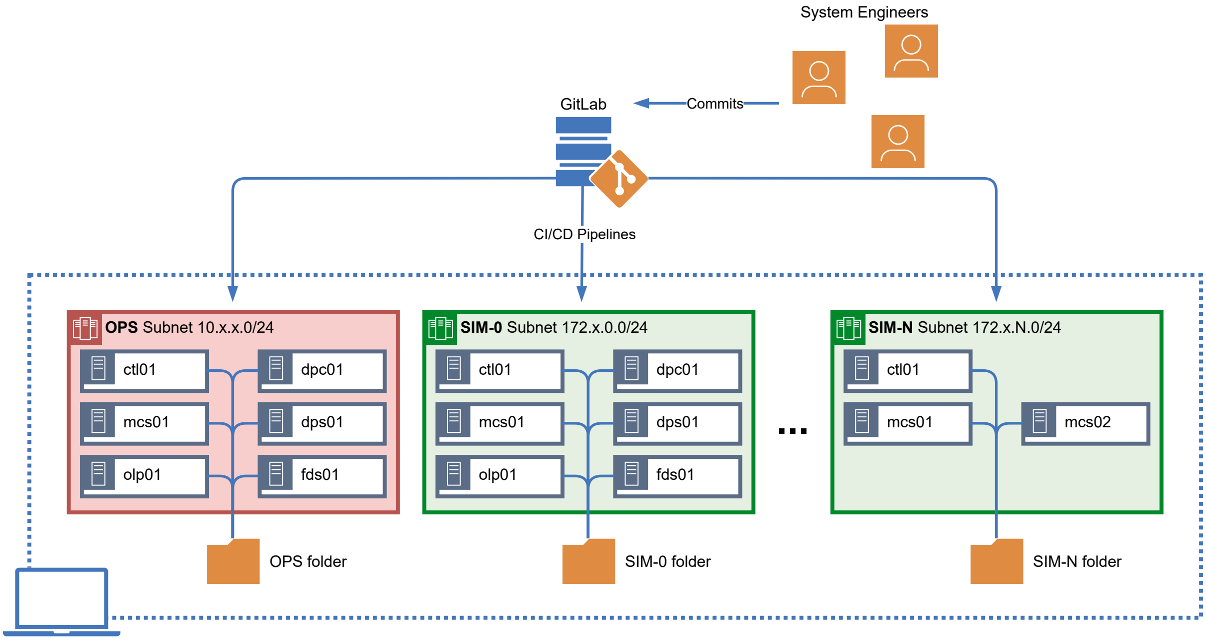


Fig. 14 Automatically managed development environment with an operational system and a variable number of simulation or development systems.

4. SARS-CoV-2 Challenges

The project faced (and still faces) some challenges due to the COVID-19 pandemic: The project officially started in October 2020 with some preliminary work being done since April 2020, i. e. the project was influenced by the pandemic and lockdown restrictions right from the start. The project team members never had the opportunity of meeting face-to-face with each other, except for two people at the same time. All work meetings were conducted

[1] https://git-scm.com/

[2] https://gitlab.com/

via teleconferences, telephone, chat systems or e-mail contact. With a general work-from-home order by DLR the project had to be setup in a way no other ground system project has been setup before. Owing to the particular concept of this project, namely designing and implementing a compact and mobile ground system, these restrictions actually turned out to be beneficial from a technical viewpoint as will be detailed in the following.

Because the project uses an agile approach, requirements engineering, system design, implementation, and tests do not happen in subsequent phases but continually and cyclically throughout the project lifetime. This means that implementation work already commenced almost immediately after project kick-off. At this point no project hardware had been procured yet, much less integrated into GSOC's network landscape. As laid out in Section 3.1 the ground system shall be largely independent of the hosting hardware. Therefore, a development environment was set up on each engineer's computer which they used to work from home. The development environment allowed each engineer to work with the complete ground system as implemented so far, which was executed on his or her own computer without resorting to any GSOC control center infrastructure. The same "infrastructure-as-code" techniques were used for the home development environments as are now used for the development environment on the project hardware. All ground system infrastructure was defined in a common Git repository right from the start, made available on each developer's machine, and designed in a robust way to overcome the differences between the developer machines. These were not only differences in hardware specifications, like the amount of available memory, but also in the software environment, like different operating systems. By being forced to make available the development environment not on one centrally accessible system but on each developer system the whole environment was portable since the beginning, thereby fulfilling one of the project goals.

Once the project hardware was delivered, the same development environment was rolled out there. Because the network setup was not yet ready at this moment, the hardware was placed outside of GSOC at first, allowing remote access to all engineers. Later, the hardware was physically moved to GSOC and connected to a network accessible from home through a virtual private network. This was already a first successful test for the mobility of the system as is stated project goal. The restrictions imposed by the pandemic almost automatically aligned with some of the project goals and helped implement them right from the start.

Of course, a development system is not comparable to the final product in all respects: While at first the development system closely resembled the setup of the final product because it allowed working from home, later on a more elaborated development environment fitting into GSOC infrastructure was needed, which is described in Section 3.4. The final setup allows multiple engineers to work on the same system, which was not possible with the work-from-home solution.

5. Conclusion and Outlook

We have shown a system design and implementation of a compact and mobile ground segment on a COTS laptop for small satellites in low-Earth orbit. It can be put to use anywhere in the world with minimal connectivity required except for a ground station, which could be equally compact and mobile in principle. Compromises in mission design due to intermittent connectivity have been assessed. Two first demonstration campaigns are currently in preparation, showing the feasibility of this concept with the optical imaging payload of the BIROS satellite. How to set up and roll out such a system in an automatic fashion has been shown in detail, as well as how the current global pandemic influenced and even partly helped the project.

Automation functionality for fully automatic workflows initiated by a single operator will be implemented, some of it already for the second demonstration. New deployment concepts like putting the ground segment on a USB stick, making it available for download, or putting it in the cloud will be assessed, as well as the new redundancy possibilities that arise from these concepts. Combining this compact and mobile Mission Operations System with an equally compact and mobile ground station also leads to fascinating new possibilities for space operations.

Acknowledgments

Our special thanks go to Martin Wickler for initiating this project and providing core ideas to it. We sincerely appreciate the project funding provided by Wolfgang Jung. Finally, we thank the team members and supporting staff bringing this project to life.

References

[1] CCSDS, "Cross Support Concept - Part 1: Space Link Extension," , Mar. 2006. URL https://public.ccsds.org/Pubs/910x3g3.pdf, Green Book, 910.3-G-3.

[2] Reile, H., Lorenz, E., and Terzibaschian, T., "The FireBird Mission – A Scientific Mission for Earth Observation and Hot

SpotDetection," *Digest of the 9th International Symposium of the International Academy of Astronautics*, Wissenschaft und Technik Verlag, 2013. URL https://elib.dlr.de/83866/.

[3] Stangl, C., Lotko, B., Geyer, M. P., Oswald, M., and Braun, A., "GECCOS – the new Monitoring and Control System at DLR-GSOC for Space Operations, based on SCOS-2000," *SpaceOps Conferences*, American Institute of Aeronautics and Astronautics, 2014. doi:10.2514/6.2014-1602, URL http://dx.doi.org/10.2514/6.2014-1602.

[4] ECSS, "Ground systems and operations – Telemetry and telecommand packet utilization," , Jan. 2003. ECSS-E-70-41A.

[5] Peat, C., and Hofmann, H., "SATMON - A Generic User Interface for Satellite Control," *SpaceOps Conferences*, American Institute of Aeronautics and Astronautics, 2004. doi:10.2514/6.2004-316-163, URL http://dx.doi.org/10.2514/6.2004-316-163.

[6] Beck, T., Schlag, L., and Hamacher, J. P., "ProToS: Next Generation Procedure Tool Suite for Creation, Execution and Automation of Flight Control Procedures," *SpaceOps Conferences*, American Institute of Aeronautics and Astronautics, 2016. doi:10.2514/6.2016-2374, URL http://dx.doi.org/10.2514/6.2016-2374.

[7] Hamacher, J. P., and Beck, T., "ProToS: Automation of Flight Control Procedures for the European Data Relay System," *SpaceOps Conferences*, American Institute of Aeronautics and Astronautics, 2018. doi:10.2514/6.2018-2541, URL http://dx.doi.org/10.2514/6.2018-2541.

[8] Wörle, M. T., Spörl, A. K., Hartung, J. H., Lenzen, C., and Mrowka, F., "The Mission Planning System for the Firebird Spacecraft Constellation," *SpaceOps Conferences*, American Institute of Aeronautics and Astronautics, 2016. doi:10.2514/6.2016-2621, URL http://dx.doi.org/10.2514/6.2016-2621.