European Space Agency

Robotic Digital Twin

Reference: ESR – TRA-ROBDT-ESR

Date: 03/11/2022

Version: V1.0

ROBDT Executive Summary Report

Contacts

Dr. Konstantinos Kapellos

Project Manager
t. +32(0)2 893 15 10
f. +32(0)4 249 68 22
e. Konstantinos.Kapellos@trasysinternational.com

Mr. Patrick Vanhoutte
Project Manager, BE DEP CENTRAL PROPOSAL & SALES SUPPORT
t. +32(0)2 893 18 17
f. +32(0)4 249 68 22
e. Patrick.Vanhoutte@trasysinternational.com

Network Research Belgium SA (NRB)
Business Unit Trasys International
Interleuvenlaan 10
3001 Heverlee
Belgium
tel. +32 (0)2 893 12 11
Registration number: 0430502430
VAT: BE 0430502430
www.trasysinternational.com

ISO 9001:2015 Quality Management — FS 706532
ISO/IEC 27001 Information Security Management — IS 706533

Status information

Contract:	ROBDT
Contract Number	4000135373/21/D/SR
Ref.:	ESR – TRA_ROBDT_ESR

Security classification:	TRASYS Confidential	State:	Final
Current version number:	V1.0	Date of first issue:	03/11/2022
Prepared by:	K. Kapellos	Date:	03/11/2022
Verified by:	I. Oikonomou	Date:	03/11/2022
Approved by:	K. Kapellos	Date:	03/11/2022

Circulation / distribution list

Name	Address	I/A	Name	Address	I/A
ESOC					
TRASYS ROBDT Team					
FBK ROBDT Team					
GMV ROBDT Team					

(I = for information, A = for action)

Table of contents

1 SIMROB Executive Summary Report

1.1 Introduction

The ROBDT project designed and demonstrated a Robotic Digital Twin framework that combines data-driven models, physics-based and symbolic models and uses online data and data analytics to adapt the models at runtime. The digital twin supports the robotic asset operations by providing timing and reliable prediction, by supporting what-if analysis to assess multiple scenarios, by identifying and diagnosing faults during operations, and finally, by detecting new science opportunities on acquired images. The "Robotic Digital Twin" Activity is funded by ESA and led by TRASYS in collaboration with FBK and GMV under the contract 4000135373/21/D/SR.

1.2 Study objectives

In robotics, Digital Models of the target systems are traditionally used in all phases of a mission under the name 'Virtual Flight Segment' ranging from design to the development and the operations. These digital systems struggle to fully support their objectives in particular when the involved models are not able to capture the complete physical reality. This is particularly true when the operations environment evolves during the mission (e.g., when discovering a new planetary area).

The ROBDT activity proposes a new framework (see Figure 1-1) where engineering methods and AI techniques are integrated into a coherent Robotic Digital Twin Framework in order to allow:

> **On-line update of the system models**: The appropriate combination of data-driven and physics-based simulation models enables the application of online data analytics for adapting at runtime the models of the virtual asset guaranteeing a high-fidelity representation of the physical asset and its environment.

> **Planning and what-if analyses**: A digital twin enables planning of actions and what-if analyses based on more reliable models. These analyses allow to synthesize unexpected scenarios and study the response of the system as well as the corresponding mitigation strategies. This kind of analysis without jeopardizing the real asset is only possible via a digital twin.

> **Plan monitoring and fault diagnosis**: Telemetry data are monitored to detect and identify anomalies. Diagnosis is performed to enable a retrospective analysis to extract the root causes of the observed failures. This is essential in order to support timely recovery from problematic situations and/or safely operate the real asset in a degraded mode of operation.

> **Operations support using AI/ML models**: Images are monitored on-line to identifies areas of scientific interest based on a 'scientific agent' including an AI/ML model trained by labelled images.

In the following paragraphs, we present the proposed software architecture to build the ROBDT system and the corresponding functionalities, with particular attention to the interplay and synergies between engineering methods, symbolic and data-driven AI. E.g., in a planetary exploration mission, the wheel-terrain interaction model used for planning is the same as the one used for simulation, and it is adapted with machine learning algorithms based on the telemetry data. Finally, we describe the proof-of-concept demonstrator based on the ExoMars planetary exploration mission.

This report is organized as follows: Section 1.3 discusses the related work; Section 1.4 details the framework's components; Section 1.5 describes the proof-of-concept demonstrator; Section 1.6 draws the conclusions; and Section 1.7 includes related bibliography.

1.3 The ROBDT framework

1.3.1 System Architecture

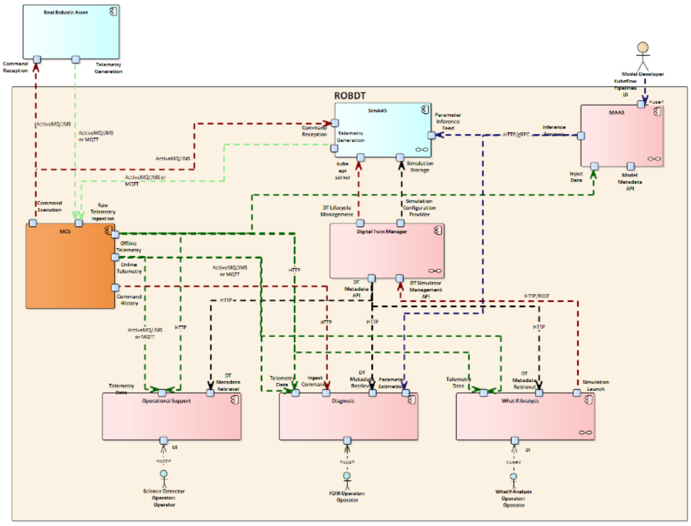

Figure 1-1: The ROBDT high-level architecture

Figure 1-1 shows the ROBDT architecture as a first layer decomposition of the software into the following components:

- **Simulation As A Service (SimAAS)**: consists of multiple simulators which simulate the functionality of the robotic assets as well as the mechanisms for their upload to the system, their configuration and management (start, stop, delete), and finally the monitoring of their status. It involves the use of Kubernetes and associated services.
- **Model As A Service (MAAS)**: provides ML Ops capabilities to the ROBDT by facilitating tasks such as data preparation, model training and model serving, while also enabling easy, repeatable, portable deployments on diverse infrastructure. It is based on the Kubeflow open-source project. In the demonstrator, two such models and the corresponding pipelines are proposed: the wheel-terrain interaction model and the Data Handling Subsystem (DHS) update model.
- **Digital Twin Manager**: the models are handled by a Digital Twin Manager (DTM), which manages DT definitions and facilitates operations associated with launching, monitoring and stopping the corresponding simulations by hiding the complexities of Kubernetes' APIs that are used in SimAAS to perform the same operations.
- **What-if Analysis (WIA)**: allows simulating the system from its current state or from a hypothetical state according to a given scenario with the additional possibility to check whether a certain goal condition is satisfied, or it is violated. In the context of this activity the WIA component focuses on the automatic activity plan generation under various initial conditions with the additional capability of rehearsing them using the available AI/ML models for reliable resources estimation.
- **Diagnosis (FDIRPM)**: allows detecting faults in the current execution or on historic data, identifying causes of the faults based on their models, and providing the corresponding feedback to the operators. In the context of this activity, we propose to focus on the detection of faults during the execution of an activity plan and to allow the operator to plan a recovery action by generating an alternative activity plan.
- **Operations Support (SCIDET)**: supports engineering or science operations planning and assessment. In the ROBDT activity, a "scientific agent" is integrated to detect predefined patterns of interest or novelty on on-line or historical images acquired by the robotic asset.
- **MCS**: for monitoring and controlling the robotic asset as well as the simulator. The MCS is complemented by the 3DROCS robotic ground control station to support specific robotic functionality.
- **Robotic Asset**: it is the physical robotic system that is under monitoring and control.

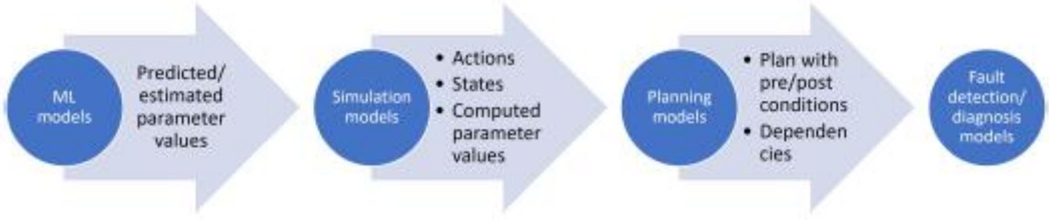

Figure 1-2: Information exchanged between ROBDT models

These components use various heterogeneous models and there is a strong interplay between data-driven models, physics-based, and symbolic models, as summarized in Figure 1-2. In the following sections, we describe in more detail the components using these models.

1.3.2 Simulator

The simulation capabilities are provided by the instantiation of the SIMROB multi-asset space robotics simulator [12]. A high-level breakdown in models of SIMROB is depicted in Figure 1-3. It includes:

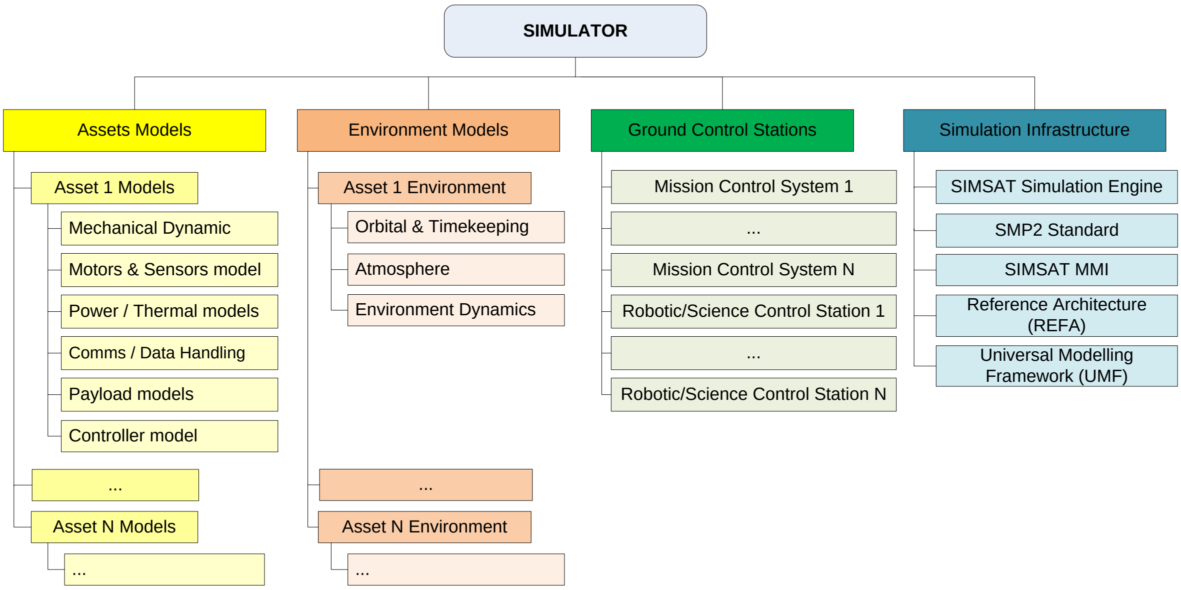

Figure 1-3: SIMROB simulator breakdown

- **The Assets models** that allow to compute the state evolution of the subsystems of the assets that are under control. For each asset, it mainly includes models of its mechanical, electrical and thermal dynamics, of its data handling, communications and payload subsystems as well as the model of its control software.
- **The Environment models** are in charge of the simulation of the environments that interact with the assets. These models encompass the reproduction of planetary and orbiter ephemerides, the provision of planetary atmospheric data as well as the morphology and the topology of the environment in which the assets evolve. In particular:
- The **Simulation framework** is based on SIMULUS. It covers both the execution of a spacecraft simulation and the simulator architectural design. The SIMSAT Simulation Engine is the 'engine' of the simulator, the SIMSAT Man-Machine Interface provides the interface between the user and the simulator, the Generic Models (GENM) comprise a suite of reusable generic simulation models, the SMP2 standard enables reuse and portability of the simulation models, the Reference Architecture (REFA) establishes a suitable breakdown of simulators into models and finally the Universal Modelling Framework (UMF) supports an efficient and smooth approach of software development for SMP2 simulations.

1.3.3 Model updaters

In the ROBDT architecture, the Model As A Service (MAAS) is a component whose goal is to provide ML Ops capabilities to the system. The MAAS facilitates data preparation, model training, and model serving and enables easy, repeatable, portable deployments on diverse infrastructure. Serving the models from MAAS allows specific digital twin components to access the model predictions for different use such as simulation and diagnosis.

Infrastructure The MAAS is based on a specific component of the Kubeflow open-source project: Kubeflow Pipeline. Kubeflow is a service capable of making deployments of machine learning (ML) workflows on Kubernetes simple, portable and scalable. The Kubeflow Pipeline is a platform in Kubeflow that has a UI and an API for triggering and tracking experiments, jobs, and runs. It also contains the engine used to manage and execute the ML workflows created by the Model Training Pipeline steps. The Kubeflow Pipelines is a platform for building and deploying portable, scalable machine learning (ML) workflows based on Docker containers. The role of the ROBDT's private Docker Registry is to manage the Docker images into which the various steps of model training pipelines are packaged. Based on Seldon's Core, the Model Server exposes trained models as web APIs, providing its clients the ability to make predictions and get information on the models. The MAAS offers interaction with the Monitoring and Control Station (MCS) component to obtain the historical and real-time telemetries. Those data are requested from the model updaters and used to build the training dataset.

Model Updater Component Based on the functionality provided by Kubeflow it is possible to trigger the training of a model manually or automatically at regular intervals. Both can be performed and configured via Kubeflow's Pipeline UI or the corresponding API. Once the pipeline is triggered, Kubeflow Pipeline's orchestration engine starts executing the described workflow. The pipeline is made up of Docker images related to each other as a graph through input and output files dependencies. The pipeline steps are the following:

- *Telemetry acquisition*: the first step of the pipeline is responsible for the telemetry acquisition. The operation is done by contacting the MCS component by REST API, which provides the requested data in JSON format.
- *Data pre-processing*: the historical telemetries of the mission are pre-processed for the training and testing phase of the model.
- *Training*: the training dataset is obtained from the previous step and loaded using a custom DataLoader. The model is instantiated and trained; the model parameters are saved on a dedicated, persistent volume. As a result of this step, the best performing model is available for the next steps.
- *Test*: the model runs on the testing dataset to validate the performance. The score is then passed to the next step to provide information about the behaviour of the updated model.
- *Deployment*: to enable inference service to the other ROBDT components, this final step provides the configuration for the Seldon Core platform about the model parameters and the Python's handler script.

1.3.4 Planner and what-if analysis

One of the main problems when managing a remote asset is to plan the activities to be performed ahead-of-time, because a significant time delay can hinder direct tele-commanding. For example, it is not possible to teleoperate an asset on the surface of Mars due to the communication delay that is in the order of tens of minutes. For this reason, the remote assets are equipped with autonomous executors of activity plans (mission management system) that need to be properly formulated on-ground ahead of time and uploaded before execution. Designing activity plans that are robust and achieve a desired objective is no trivial task when the complexity of the target system is more than trivial. For this reason, automated planning technologies have been historically employed in space applications. The usefulness of such technique is not limited to the generation of plans for the immediate future, but also to investigate hypothetical situations and to perform so-called "what-if analyses" (WIA): before committing to a specific plan or addressing a problematic situation, a planning and simulation system allows the study of different plans and objectives in different real or hypothetical situations. Finally, WIA allows for retrospective analyses: in light of new model information, one can re-assess past decision in order to improve future decision-making.

Within the ROBDT framework, WIA is seen as a service that takes advantage of the superior precision of digital twin models: thanks to the strong alignment between the models and the physical assets it is possible to provide more realistic estimations of the costs of a certain plan and ultimately to provide a better support for decision-making. Moreover, an interesting and pivotal feature offered by digital twins is the evolution of models, making it possible to adapt automated planning to the degradation of capabilities or to the evolving conditions of the environment.

On the practical side, the overall idea behind the digital-twin-enhanced WIA is to maintain a model-based approach for planning and high-level simulation, but to allow for parameters that are to be estimated/learned from the telemetry data within the digital twin. Concretely, we demonstrate the behaviour of automated planning when some parameters (in particular the duration of some activities and therefore the power consumption) are estimated by means of ML models. Ideally, having a more precise timing model of the system allows a less conservative planning which in turn allows to fully exploit the remote asset capabilities.

1.3.5 Fault detection and diagnosis

Diagnosing faults is essential in order to detect and identify anomalies that could endanger the real asset. To this aim, the system is equipped with a Fault Detection, Isolation, and Recovery (FDIR) component that monitors the telemetry data and the execution of the activities, in their initial, in progress and terminating phase. A relevant aspect strictly related to the monitoring phase is the ability to perform its task even in absence of complete information on the state; for example, both the state of some components and the currently executed action may be unknown. In such a case, a set of belief states that are compatible with the observations, strictly contained in the set of all possible states, is considered and the diagnosis phase can be employed with this partial information. With the aim of addressing the partial observability problem we have used NuRV; this tool is able to generate a monitor for a specific LTL property to be monitored on a given system. In this case the system consists of the plan, which can be seen as a loop-free algorithm, synthesized by the planner component, while the conditions to be monitored are pre, in progress and post conditions. In case an anomaly is detected, a retrospective analysis based on the DT models and the historical information on past states, is carried out to localize the possible faults and identify the root causes of the failure. Diagnosis is based on a fault model which describes the effects of faults, their dependencies and the fault propagation rules. The adaptation of the DT models at run-time can improve the situational awareness of the real asset and provide a more precise analysis with respect to the FDIR capabilities of the physical system alone. When an anomaly is detected, the FDIR component could trigger a reconfiguration of the system, e.g., to continue operation in a degraded mode. FDIR can also be used to aid predictive maintenance by supporting the detection of the performance degradation of some component. Finally, the FDIR component provides a service to the what-if analysis functionality, namely it supports the planning activities by monitoring the plan execution in order to detect and identify unexpected outcomes of the actions.

1.3.6 Operations support – Scientific agent

The scientific agent identifies areas of scientific interest in on-line images: During the design phase, the neural networks of the 'scientific agent' are trained by labelled images. During operations, images from the downloaded data are provided:

- Initially, interesting/salient regions are detected in the input image generating region candidates to be processed in subsequent steps; afterwards, a trained variational autoencoder is used to encode and decode the region proposal. The reconstruction error, as well as latent space losses, are used as an indication that the target is novel (unknown).
- Afterwards, a set of VAE-based trained models evaluate each region's proposal to determine its class. The classification is based on a semi-supervised approach in which convolutional neural networks (CNN) are trained on different losses (reconstruction loss, latent space loss, and mixed losses) which result from the VAE.
- Finally, the results of previous steps are consolidated and all known and/or unknown targets are reported in form bounding boxes (including a score and/or a measure of uncertainty) and locations. Outputs of the scientific detector are reported to the MCS as suggestions of areas to be explored.

The classification provided for the scientific agent trained with Public Mars novelty detection Mastcam labelled dataset (https://zenodo.org/record/1486196) around 100k images (see Figure 1-6). The Mars novelty detection Mastcam labelled dataset, consists of sub-sampled images (64×64 pixels) of the Mars Science Laboratory (MSL) Analyst's Notebook which were obtained by NASA's Curiosity rover. The Mastcam is a multi-spectral camera and captures images at different wavelengths giving each image six channels. This classification is based on a set of classes: background, broken rock, drill hole, drt, dump pile, and novelty detection.

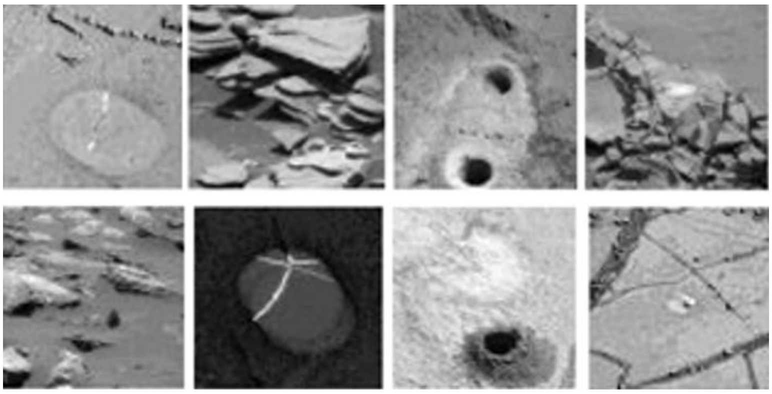

Figure 1-4: Example of Public Mars novelty detection Mastcam labelled dataset

1.4 The ROBDT proof-of-concept demonstrator

For the demonstration of the ROBDT framework, we use the ExoMars surface operations. Because of the inherent uncertainty of the robot-environment interaction, data-based models are well suited. Moreover, it is an ideal case study for path planning and monitoring. Let us consider a typical scenario prepared for a 'sol' execution from the ExoMars planetary exploration mission: the 'Drilling site approach and surface sample acquisition' (see Figure 1-8). The following activities shall be performed autonomously under the constraints of the available power, memory capacity for data storage, and duration (single sol). Initially, the rover waits the transition Night to Day to wake-up (driven by the rover PCDE, when the solar panels generate power greater than a threshold (20W)) and configures accordingly the rover for the day activities. In particular, the subsystems involved for travelling are warmed-up and moved to a 'standby' state. These steps involve several uncertainties, mainly the exact local Mars time at which the rover wakes-up as well as the warm-up durations, which all depend on the external conditions (e.g., atmospheric temperatures, relative orientation of the rover solar panels with respect to the Sun, etc.).

At completion of the rover configuration, the rover starts traveling to reach the outcrop whose position has been identified from ground. Although the duration of the travel depends on the topology and characteristics of the encountered terrain, it can be estimated at planning time. At arrival at the outcrop, the rover is unconfigured from travelling operations and configured for drilling: travel related units are switched off while the drill box and the drill are warmed-up and moved to the 'standby' state. Again, the expected durations (and therefore power consumptions) have to be estimated as they depend on the time in the 'sol' that the rover reached the outcrop. At the next step the drill box is deployed, the drill is initialised, and reaches the soil to collect the surface sample (10cm depth). Afterwards, the drill retracts. The duration of the sampling procedure, and therefore the power consumption, depends on the hardness of the soil. Finally, images of the environment shall be acquired and downloaded to guarantee that the ground planning team has enough information for planning for the next sol. After establishing the communications with the orbiter and transferring the acquired data, the rover waits the transition Day to Night (driven by the rover PCDE, when the solar panels generate power less than the threshold of 20W), configures the Rover for night and 'sleeps' waiting for the next plan to be uploaded for execution.

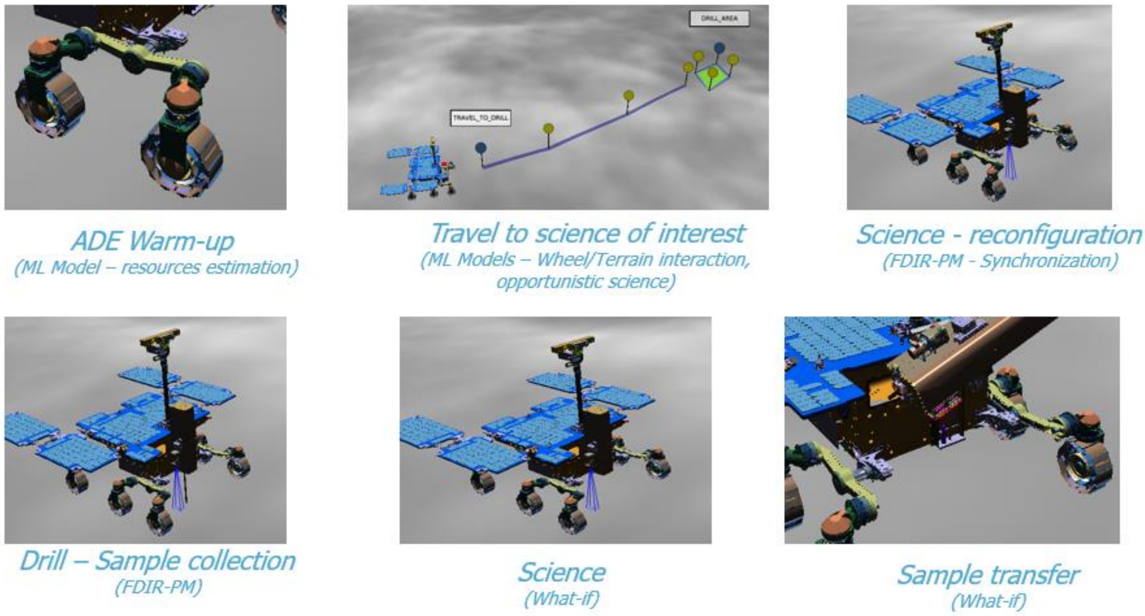

Figure 1-5: The subsurface sample acquisition main phases

In this scenario, the use case foresees:

- **For the ROBDT system**, there are two specific machine learning models that are considered: the wheel terrain interaction (WTI) model and the DHS model that predicts the Actuator Drive Electronics (ADE) warm-up time. Those elements run on top of the MAAS component taking advantage of the Kubeflow Pipeline architecture. In both cases, the approach used to fix the model architecture is the same. As an example, we present here the case of the WTI model. The purpose of the WTI model is to estimate the drawbar pull of the robotic asset and the power consumption of the motors. These calculations are made from the terrain and rover characteristics given as input from pre-processed telemetries. In particular, the inputs of the model are the terrain type (class), slip ratio (%), stiffness (%), and normal axis load (Newton). The outputs are the average drawbar pull (Newton) and the power consumption (Watt). Given the small number of input features of the model, the advantage of using Deep Learning solutions is not clear, as they usually require many parameters and are therefore heavy to train. For this reason, we are testing two different solutions to achieve the described functionality: the first is based on a Deep Learning Fully Connected Neural Network, and the second uses Boosted Decision Tree. While both can fit in the pipeline based on the MAAS, predictive power and computational resources required for the online adaptation can be different and must be carefully evaluated. The framework adopted for the development of the fully connected neural network is Pytorch. The neural network solution is composed of an initial embedding layer that maps the terrain class to a higher dimensional space. Then, this embedding is merged with the other features forming the input of the first layer of the network. A full representation of the network architecture is presented in Figure 4. For the boosted decision tree we used the Catboost Python library [29].
- **For the planning part**, we are setting up and experimenting with a fully integrated solution, which is capable of planning with semi-opaque models. In particular, we have access to a model of the activities and tasks of the rover and we need to automatically synthesize activity plans. However, some of the actions have simulated effects, meaning that the consequences of applying such actions are not modelled but can only be simulated, and the duration of some actions are also not formally modelled. Most notably, among these "evaluatable" quantities, we have the ADE warm-up timing that is estimated by a learned and evolving ML model as discussed above. We use the AIPlan4EU Unified Planning library to model simulated effects and the TAMER planner3 for the actual plan generation. The results showed that the approach is capable of generating valid plans quickly considering the quantities estimated by means of ML.
- **As for the fault detection part**, we are monitoring the execution of the plan. The telemetry provides complete information about the state components, but no information about the state of the task

execution. Looking at the case study, the plan provided by the planner requires first to run a task that waits for the rover to warm up and switch on a set of subsystems for traveling. Upon completion of this action, a state component is modified. This state component acts as a precondition for a task that is used for updating the rover heading estimate with a value provided by ground. If the monitor notices that the heading has just been changed, but the precondition state component of the heading update task has not been satisfied before, it can decide that the heading update task has been executed violating its preconditions in one of the possible belief states reports this violation to the operator and to the diagnostic component.

> Given the anomalies identified by the monitoring component, the goal of the **diagnosis component** is to provide a list of most probable explanations for these anomalies. The explanations are identified using a fault propagation graph (FPG), which describes how failures of one subsystem or component of the rover can cause failures of other components. In particular, for ROBDT, we construct the FPG as follows. First, for each task, we use the DT specification to identify the actions of other subsystems that can cause a failure of the given task. For example, the warm-up task that prepares the rover for travel depends on warming up the navigation cameras, localization cameras, actuator drive etc. Second, we use the description of hardware implementation and FMECA tables to describe how failures in the hardware components can cause failures of the higher-level subsystems. For example, the actuator drive depends on working hold-down release mechanism, which in turn depends on working motors, motor heaters, etc. We then use efficient techniques rooted in formal methods that for each set of identified failures list all the possible root causes. As a result, if the monitoring component reports an anomaly in the warmup task, we can list failure of motors as one of the root causes (among many others). More interestingly, if the monitoring component reports several anomalies, which all transitively depend on the motors, the diagnosis component can report the motor failure as the most probable root cause as it is more probable than multiple separate failures of independent subsystems.

1.5 Conclusion, Lessons Learned and Recommendations

In the ROBDT study we proposed a generic and extendable architecture of a Robotic Digital Twin. We focused on the planning, monitoring fault detection and diagnosis functionalities and the related combination of data-driven physics-based, and symbolic models. Finally, we demonstrated the architecture on the ExoMars planetary surface operations. In the process of building and using the proof-of-concept demonstrator we identified several strengths and weaknesses of the approach as well as guidelines for the mission activities modelling. In addition, a roadmap for future activities is proposed underlying the interest to deploy the proposed architecture for future missions under preparation as PROSPECT, Rosalind Franklin 2028 and Argonaute.

In the future, we are going to use the same infrastructure to validate and verify autonomous systems with AI/ML components with a simulation-based system level approach. This is part of VIVAS, another ESA-funded study started in May 2022.

END OF DOCUMENT